University of Alberta

SET COLOURING GAMES ©

by

Jack van Rijswijck

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of **Doctor of Philosophy**.

Department of Computing Science

Edmonton, Alberta
Fall 2006

Your file Votre référence
ISBN: 978-0-494-23121-0
Our file Notre référence
ISBN: 978-0-494-23121-0

Abstract

The game of Hex has been of particular interest to mathematicians, computer scientists, and game players alike ever since its discovery in the 1940s. The game is provably difficult in the sense of algorithmic complexity, yet its rich mathematical structure allows for many properties to be provable even when exact solutions to the game are unknown. Such properties can then be used to boost computational approaches to solving Hex on small boards, and playing well on larger boards.

This thesis presents a general class of mathematical games that contains Hex and many of its relatives. The thesis generalizes all previously known Hex theory to this class, and identifies conditions that give rise to these properties. This enables rigorous proofs of game properties previously known only colloquially, as well as introduction of new properties. Algorithmic optimizations that follow from this theory have enabled advances in Hex solving and playing, and can be applied to related games as well.

Acknowledgments

Special thanks go to...

my supervisor **Ryan Hayward,** for sparking both the start of this thesis, with fruitful ideas and contributions, as well as the end of it, with constant encouragement to get the thesis done with since my Hex activities will forever be a work in progress anyway;

Michael Buro, Martin Müller, Mazi Shirvani, and **Bjarne Toft** for serving on my committee;

the University of Alberta **GAMES Group,** for their support and boundless enthusiasm and fascination with games;

Cameron Browne, for his Hex drawing macros, and **Axel Ostmann** and **Yōhei Yamasaki** for helping me with their papers;

Gábor Melis, for developing an exceptionally strong Hex playing program and being very open and helpful in discussions and exchanges of ideas;

Edith Drummond at the Computer Science administration for navigating me through years of paperwork and putting up with my chaotic coordination;

and finally, special thanks to everyone who was patient with me while I asymptotically finished this thesis; in particular my colleagues and managers at **Google**, and, most of all, my parents **Pieter and José**.

Contents

Chapter 1

Introduction and Motivation

> Suddenly in the half-light of dawn a game awoke, demanding to be born.
>
> Today it is ready for release into the world and that is what I, in this Christmassy innocence, shall attempt.
>
> Ideas such as this carry a certain romantic note as if they were gifts from above, fruits of mystical inspiration. The truth is that you are given them seemingly free of charge after having toiled seemingly in vain."
>
> Piet Hein [50, 64].

1.1 Hex Appeal

Though this thesis deals with a general class of abstract games, its main motivation lies in the ongoing study, both theoretical and practical, of the game of Hex. The Danish mathematician, poet, and engineer Piet Hein first introduced Hex – which he named "Polygon" – in 1942, during a lecture at the University of Copenhagen, and in a series of columns in the newspaper Politiken [51]. In 1947 John Nash independently came up with the same game at Princeton, where it become known as "Nash".[1]

Hex has an unmistakable appeal to mathematicians, as evident from Hein's quote, and also from:

> Nash [Hex] was originally discovered in Denmark, and rediscovered by the author at Princeton. – John Nash [71].

[1] Of the popular story that it was also named "John" [34], Nash writes: "It is not true that the idea came from an actual bathroom floor [at Princeton], but the concept of an hexagonally tiled bathroom floor was *talked about* among the grad students at that time and I think the name 'John' was thought of or joked about at that time also." [72]

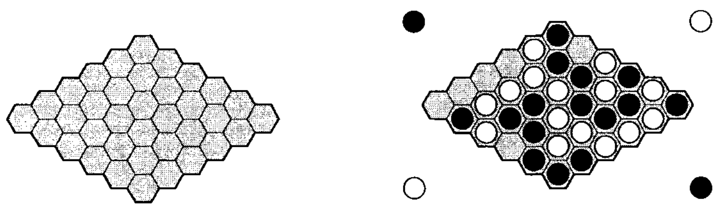


Figure 1.1: An empty 6×6 Hex board (left) and a completed game with a win for White (right)

> We do not pretend that Hex was the invention of the Emperor Shun, who reigned over China almost three thousand years ago, and who thus wished to revive the wavering intelligence in his son Shang Kiun; nor that the Hebrew, Persian, Egyptian, or Hindu sovereigns, impassioned by astronomy and strategic ideas, practised it before our times. No, contrary to the games of Go and chess, Hex was discovered in the middle of the 20th century. – Claude Berge [11].

The recurring theme is that Hex is often viewed as a *discovery* rather than an *invention*, having a Platonic existence of its own.

The game's appeal lies in the contrast between the simplicity of its rules and the subtlety of its play. It is quite literally "a minute to learn, a lifetime to master". Hex has many mathematical connections, and much can be proved about the game. In addition, Hex is also compelling from a computational point of view. The game is scalable with its board size, and polynomial solutions likely do not exist.[2] Despite this – or indeed *because* of it – large reductions can be achieved in the computing time necessary to solve small-scale Hex problems, and to find heuristically strong moves for larger scale boards.

1.2 Hex Rules

The game of Hex is played on a board containing a rhombic array of hexagonal cells with an equal number of hexagons on each side,[3] as in Figure 1.1. A commonly used board size is 11×11, but any size can be used.

The two players, Black and White, take turns placing a piece on the board. White's objective is to form a chain of adjacent white pieces connecting the lower-left side of the board to the upper-right side. Black's goal is to connect the upper-left side to the lower-right-side with a black chain. Lest the players forget which two sides they are trying to connect, a pair of extra pieces has been placed off the board to remind them. Figure 1.1 shows a completed game in which White has won.

In practice, the first player to move has a significant advantage. This is often offset by using the

[2] See Section 3.5.

[3] If the side lengths are unequal, there is a trivial winning strategy for the player to traverse the shorter distance [34].

swap rule: one player places a piece of any colour anywhere on the board, and the second player then chooses which colour of pieces to adopt. The game continues with the next move played by whomever owns the colour opposite of the colour of the first piece played. This "I cut, you choose" convention will deter the first player from playing an opening move that is particularly advantageous to either colour.

These simple game rules lead to deep strategic subtleties. For an excellent overview of Hex strategy, see [18].

1.3 Motivation

Certain strategic properties of Hex were known immediately from its discovery in the 1940s. However, Hex was also the first "natural" game to be proved PSPACE-complete.[4] Thus the game is not in NP, which means that solutions to Hex problems cannot generally be verified with a polynomial amount of computation. For this reason, these strategic properties tend to be non-constructive.

The following list gives a flavour of the known Hex theorems, such that the reader may anticipate the purpose behind the results presented in Part 1.

1. [Hein, 1942; Nash, 1947] On any board size there exists a winning opening move.
2. [Hein, 1942; Nash, 1947] Adding a friendly piece or removing an enemy piece is never disadvantageous.
3. [Beck, 1969] On any board size there exists a losing opening move [10]. See Figure 1.2 for two opening moves that Beck proved to be a loss, as well as a proven losing reply to a losing move.
4. [Schensted and Titus, 1975] Any move that is surrounded by only three regions, and many moves that are surrounded by four or five regions, should be avoided [91]. See Figure 1.3.
5. [Hayward, 2003] Any move on the second row dominates the two underlying moves on the first row. Stronger still: after a move on the second row, the two underlying cells on the first row can be "filled in" [44]. See Figure 1.4.
6. In Figure 1.5, both players should avoid move x.

Hayward's theorem proved of key importance in determining the outcome with optimal play for all opening moves on the 7×7 board [46]. The last theorem is by the author. It inspired a gradual uncovering of a more general theory, of which all the mentioned theorems are special cases. This addresses a wish by Claude Berge, mathematician and avid Hex player:

> It would be nice to solve some Hex problem by using nontrivial theorems about combinatorial properties of sets (the sets considered are groups of critical [cells]). It is not

[4] See Section 3.5.

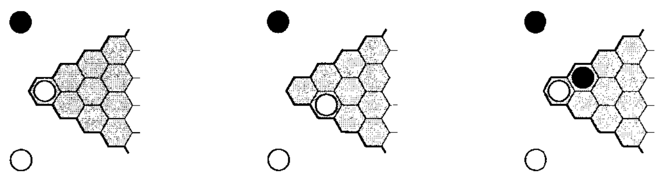


Figure 1.2: Beck's theorems: Losing opening moves for White (left, middle) and a losing reply for Black (right) on any board size.

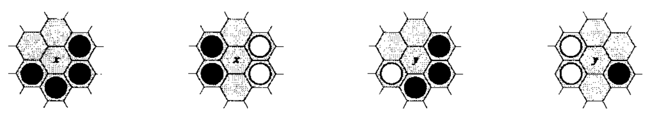


Figure 1.3: Schensted's theorems: moves marked 'x' should be avoided by both players; moves marked 'y' should be avoided by White.

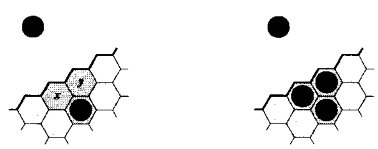


Figure 1.4: Hayward's theorems: Black's second-row move dominates the first-row moves x and y underneath (left); the position is strategically equivalent to the position on the right.

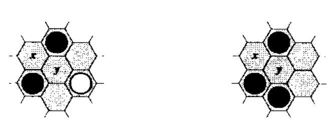


Figure 1.5: Both players should avoid move y.

> possible to forget that a famous chess problem of Sam Lloyd (the "comet"),[5] involving parity, is easy to solve for a mathematician aware of the König theorem about bipartite graphs; also in chess, the theory of conjugate squares of Marcel Duchamp and Alberstadt is a beautiful application of the algebraic theory of graph isomorphism (the two graphs are defined by the moves of the kings). – Claude Berge [12].

Indeed the theory extends to a much more general class of games, namely the *set colouring games* to be described in Part I.

The assertion that on any board size there exists a winning strategy for the first player was recognized intuitively by Hein and Nash. In fact Nash specifically composed Hex as an example of a non-constructive proof of a winning strategy. The proof as it is usually presented involves the "strategy stealing argument", but a bit of handwaving is required to gloss over some assertions that are themselves intuitively obvious, notably the theorem about adding and removing pieces. This thesis offers the first rigorous proof of these theorems, extended to the more general class of set colouring games to make it more worthwhile.

Applications of the theory lead to algorithmic optimizations that enable Hex positions to be solved on larger boards than was previously possible. The state of the art for exact results in Hex was that the 6×6 opening position was solved by Enderton in the 1990s [28], a library of solved 6×6 opening positions was compiled by the author in 1999-2002 [81], some 7×7 and one 8×8 opening move were solved by Yang in 2002-2003 [106, 103], and all the 7×7 opening moves were solved by Hayward *et al.* in 2003 [46]. The pattern of winning and losing opening moves on various board sizes appears progressively less intricate as the board size increases to 6×6, but then becomes unexpectedly irregular on 7×7, leading to the question of what the 8×8 pattern might look like.

As well, applications of the set colouring games theory lead to a higher heuristic level of play on board sizes too large to solve perfectly. The state of the art was dominated by the author's program *Queenbee* in the 1990s [83] and then by Anshelevich's *Hexy* in 2000-2002 [6] and Melis' *Six* from 2002 on [68, 69]. According to its author, Six's play rivals but does not surpass top human play on the 11×11 board [67]. The balance of strength shifts from the computer's favour on small boards to the human's favour on large boards; as of 2004, the crossover point would approximately be at 9×9.

1.4 Overview

The first part of this thesis involves the theory of set colouring games, while the second part focuses on computational aspects.

[5] Lloyd's chess problem, now considerably less famous, is reproduced in Appendix A.3.

Part I: Theory

Set colouring games are a natural extension of GAME-SAT, which is itself a natural generalization of Hex and a number of other games. GAME-SAT was introduced by Zhao and Müller [107], though games on Boolean formulas had been studied before.[6] The treatment of set colouring games in terms of game transformations is new. The chapters are organized as follows:

Chapter 2: Definitions. Conventions for notations and the naming of variables and constants.

Chapter 3: Set Colouring Games. Definition of set colouring games, strategic concepts, and strategy theorems.

Chapter 4: Related Games. Games that are special cases of set colouring games, and classes of games that cannot be modelled as a set colouring game.

Chapter 5: Minimax Values. The minimax function and its behaviour for various subclasses of games and moves.

Chapter 6: Metagames. Combining several games into a new games, and its opposite, namely decomposing a given game.

Chapter 7: Combinatorial Game Theory. A new extension of CGT to cover binary games, which are combinatorial games that end in a binary value and whose winning criterion does not depend on the last player to move.

Chapter 8: Superrational Play. A theory that enables some moves to be proved irrelevant based purely on local considerations, even in an inherently global game.

Chapter 9: Dynamic Traces. Automatically discoverable patterns that can prove the value of a position by considering only the "relevant" part of the game board.

Any theorems that appear in the text will be stated without proofs for reasons of brevity and legibility. The proofs will then be supplied in the last section of the chapter in question.

Part II: Computation

The game transformation theorems can be used to increase algorithmic performance by reducing the size of the search space by orders of magnitude. This increases the board size on which perfect solutions for Hex can be computed within reasonable resource limits, as well as the level of play on larger board sizes. Based on viewing Hex as a special case of other games, new methods can be deduced for heuristic evaluation of board positions. These heuristics are important both for play on large board sizes, as well as optimizing the search effort for exact solutions on smaller board sizes by guiding the search in favourable directions.

[6] See Section 4.2.

Chapter 10: Properties of Hex. Known theorems proved by the new methods introduced in Part I, and statistics on the description of Hex as a set colouring game.

Chapter 11: Artificial Intelligence Approaches. Algorithmic techniques known from literature for solving games and for heuristic play in abstract games.

Chapter 12: Shannon Game Heuristics. Methods known from literature specific to Hex, and its direct generalization, the Shannon game.

Chapter 13: Dead Cell Analysis. The application of superrational play to the game of Hex.

The thesis is concluded by a discussion and list of open questions and future work in **Chapter 14.**

1.5 Goals

The theory in Part I was inspired by the discovery of "win patterns", "dead cells", "fill-in", and "killing moves" in Hex. The latter three concepts, exemplified in Theorems 5 and 6 of Section 1.3, were then generalized to the multi-Shannon game,[7] which led to the development of set colouring games. Attempts to apply combinatorial game theory to set colouring games consisting of independent components spurred the development of a theory of binary combinatorial games.

The goals for Part I of this thesis relate to the theory of set colouring games:

- establish a theory and notation system for set colouring games;
- generalize the notion of "win patterns" to set colouring games;
- generalize the notion of "dead cells", "fill-in", and "killing moves" to set colouring games;
- develop a theory enabling the combinatorial analysis of set colouring games consisting of independent components.

The goals for Part II involve Hex in particular, and the practical computational aspects of the theory results for artificial intelligence approaches:

- prove previously known Hex theorems using the theory and language of set colouring games;
- show that these previously known theorems are special cases of more general theorems, applying to a more general class of games;
- derive search algorithms from the set colouring games theory, and implement them for Hex in particular and GAME-SAT[8] in general;

[7] see Section 13.6.
[8] See Section 4.2.

- test the potential usefulness of the methods in the field of QBF[9] solving;
- derive new heuristic evaluation methods for set colouring games and Hex;
- build an opening library of solved 6×6 and 7×7 Hex positions;
- solve the 8×8 Hex opening position;
- improve the standard of heuristic play for Hex on 11×11 boards.

1.6 Contributions and Publications

Following the goals outlined in the previous section, the contributions from this thesis can be summed up as follows:

- introduction and development of a theory and notation system for set colouring games;
- proof of standard game-theoretical results for set colouring games;
- generalization of "win patterns" to the concept of dynamic traces;
- generalization of "dead cells", "fill-in", and "killing moves" to the concept of captured and dominated sets;
- generalization of multi-Shannon strategy to the concept of superrational play;
- introduction of a theory of binary combinatorial games.

This realizes all the goals set out for Part I. Some of the goals for Part II are still open. Realized goals for Part II are:

- rigorous proofs of existing and new Hex theorems using the theory of set colouring games;
- completion and confirmation of the ideas of multi-Shannon strategy as instances of superrational play;
- derivation of search algorithms based on dynamic traces and evaluation algorithms based on superrational play;
- derivation and theoretical justification of search and evaluation heuristics based on Monte Carlo analysis;
- computation of an opening library for 6×6 Hex and some 7×7 opening positions.

[9] See Sections 4.1 and 11.4.

The dynamic trace and superrational play methods were a crucial ingredient in the solution to the 7×7 opening position [46] and have been incorporated in the state of the art heuristic Hex playing programs *Queenbee* and *Mongoose*.[10]

The following papers have appeared or been accepted for publication as part of this research:

[52] H. J. van den Herik, J. Uiterwijk, and J. van Rijswijck. Games Solved: Now and in the Future. *Artificial Intelligence*, 134(1–2):277–312, January 2002.

[84] J. van Rijswijck. Search and Evaluation in Hex. Technical report, University of Alberta, 2002.

[46] R. Hayward, Y. Björnsson, M. Johanson, M. Kan, N. Po, and J. van Rijswijck. Solving 7×7 Hex: Virtual Connections and Game State Reduction. In H. J. van den Herik, H. Iida, and E. A. Heinz, editors, *Advances in Computer Games ACG-10*, pages 261–278. Kluwer Academic Publishers, Boston, 2003.

[47] R. Hayward, Y. Björnsson, M. Johanson, M. Kan, N. Po, and J. van Rijswijck. Solving 7×7 Hex with Domination, Fill-In, and Virtual Connections. *Theoretical Computer Science*, 349:123–139, 2005.

[49] R. Hayward, J. van Rijswijck, Y. Björnsson, and M. Johanson. Dead Cell Analysis in Hex and the Shannon Game. In *Graph Theory 2004: In Memory of Claude Berge*. Birkhauser, 2005.

[48] R. Hayward and J. van Rijswijck. Hex and Combinatorics. *Discrete Mathematics*, to appear, 2006.

[85] J. van Rijswijck. Binary Combinatorial Games. In Richard Nowakowski, editor, *Games of No Chance 3*. To appear, 2006.

[10]See Section 11.5.

Part I

Theory

Chapter 2

Definitions

Throughout the text, constants and variables shall consistently be indicated with the same symbols. Refer to Appendix A.1 for a list of the notation conventions. All operators and functions are summarized in Appendix A.2.

2.1 Sets and Colourings

For a set $\mathcal{S}$ and an element v, let $\mathcal{S} + v$ refer to $\mathcal{S} \cup \{v\}$ and let $\mathcal{S} - v$ refer to $\mathcal{S} \setminus \{v\}$. The empty set is denoted with $\varnothing$. The notation $v, w \in \mathcal{S}$ is shorthand for $v \in \mathcal{S} \wedge w \in \mathcal{S}$. The powerset $2^{\mathcal{S}}$ is the set of all subsets of $\mathcal{S}$. A family $\mathcal{F} \subseteq 2^{\mathcal{S}}$ of subsets of $\mathcal{S}$ forms **a partition** of $\mathcal{S}$ if the members of $\mathcal{F}$ are pairwise disjoint, and their union is $\mathcal{S}$. Thus every element of $\mathcal{S}$ is contained in exactly one of the members of $\mathcal{F}$.[1]

The set $\mathbb{N}$ is the set of all non-negative integers, thus including 0, and $\mathbb{Z}_n$ is the set of the first n integers: $\mathbb{Z}_n \stackrel{\text{def}}{=} \{0, 1, 2, \ldots, n-1\}$. The set $\mathbb{B}$ is the set of the two **Boolean values**. These values will be represented as FALSE and TRUE, as well as numerically with the values -1 and $+1$, respectively. For $t \in \mathbb{B}$, the notation $-t$ refers to t's negation, so that $\{t, -t\} = \mathbb{B}$. The set $\mathbb{T}$ is the set of the three **Ternary values**. The symbol ϕ is used to represent the value "undecided", so that $\mathbb{T} \stackrel{\text{def}}{=} \mathbb{B} + \phi$. Numerically, ϕ will be represented as 0.[2] A set X is **pure** if it does not contain ϕ; the notation $\breve{X} \stackrel{\text{def}}{=} X - \phi$ indicates the "purified version" of X.

Definition 2.1.1. Let $\mathcal{S}$ and X be two sets, where X represents a set of **colours**. Then any function $\psi : \mathcal{S} \to X$ is **a colouring** of $\mathcal{S}$. An element v is **coloured** in ψ if $\psi(v) \neq \phi$, and **uncoloured** otherwise. A colouring is **complete** if it contains no uncoloured elements, and **incomplete** otherwise.

[1] Note that this does not require that all members of $\mathcal{F}$ be nonempty.

[2] In some other texts the values for FALSE, TRUE, and ϕ are represented as 0, 1, and 2, respectively, in $\mathbb{Z}_3$. For the purposes of this thesis, the chosen representation is much more convenient.

The following terms and notations are used:

- if $\psi : \mathcal{S} \to X$ then $\mathcal{D}(\psi) \stackrel{\text{def}}{=\!=} \mathcal{S}$, the set $\mathcal{D}(\psi)$ is called the domain of ψ;
- $\psi_v \stackrel{\text{def}}{=\!=} \psi(v)$ for $v \in \mathcal{S}$;
- if $\mathcal{S} = \mathbb{Z}_n$ then ψ can be represented by the vector $(\psi_0, \psi_1, \ldots, \psi_{n-1})$;
- $\psi^{-1}(\chi) \stackrel{\text{def}}{=\!=} \{v \in \mathcal{S} | \psi_v = \chi\}$.
- $\chi^{\mathcal{S}}$ is the function that maps everyy element of $\mathcal{S}$ to χ: $\chi^{\mathcal{S}} \stackrel{\text{def}}{=\!=} \xi \mapsto \chi$ for $\chi \in X$;
- $\psi \subseteq \psi'$ when $\mathcal{D}(\psi) \subseteq \mathcal{D}(\psi')$ and $\psi_v = \psi'_v$ for all $v \in \mathcal{D}(\psi)$;
- if there is a partial order defined on X, then $\psi \geqslant \psi' \stackrel{\text{def}}{\Longleftrightarrow} \forall_{v \in \mathcal{S}}[\psi_v \geqslant \psi'_v]$.

In particular, if $X = \mathbb{T}$ then $\text{T}^{\mathcal{S}}$, $\text{F}^{\mathcal{S}}$, and $\phi^{\mathcal{S}}$ will be used for the colourings that assign the values TRUE, FALSE, and ϕ, respectively, to all elements of $\mathcal{S}$.

Definition 2.1.2. The set $X^{\mathcal{S}}$ denotes the collection of all colourings of $\mathcal{S}$ with colours from X.[3] Such a set $X^{\mathcal{S}}$ is **a colour space**. If $\mathcal{S} = \mathbb{Z}_n$ then the colour space will be indicated with X^n rather than $X^{\mathbb{Z}_n}$ for clarity.

Definition 2.1.3. Consider a set of colours X with $\phi \in X$. Define the function $\text{PROJ}_{\mathcal{S} \to \mathcal{S}'} : X^{\mathcal{S}} \to X^{\mathcal{S}'}$ as follows for $\psi \in X^{\mathcal{S}}$:

$$\text{PROJ}_{\mathcal{S} \to \mathcal{S}'}(\psi) : \xi \mapsto \begin{cases} \psi(\xi) & \text{if} \quad \xi \in \mathcal{S}, \\ \phi & \text{if} \quad \xi \notin \mathcal{S}. \end{cases}$$

This colouring is the **projection** of ψ onto $\mathcal{S}'$. The notation $\psi \searrow \mathcal{S}'$ is used as shorthand for $\text{PROJ}_{\mathcal{D}(\psi) \to \mathcal{S}'}(\psi)$. For colourings ψ and ψ', the notation $\psi' \subseteq \psi$ indicates that $\mathcal{D}(\psi') \subseteq \mathcal{D}(\psi)$ and $\psi' = \psi \searrow \mathcal{D}(\psi')$.

2.2 Re-Colouring

Any two colourings can be combined to form a new colouring by re-colouring the elements that they have in common.

Definition 2.2.1. Let $\psi \in X^{\mathcal{S}}$ and $\psi' \in X^{\mathcal{S}'}$. The colouring $\psi\psi' : \mathcal{S} \cup \mathcal{S}' \to X$ is defined as:

$$\psi\psi' : \xi \mapsto \begin{cases} \psi'_\xi & \text{if} \quad \xi \in \mathcal{S}', \\ \psi_\xi & \text{otherwise.} \end{cases}$$

[3]The powerset of $\mathcal{S}$ can be seen as the set of all Boolean colourings of $\mathcal{S}$, so that $2^{\mathcal{S}} = \mathbb{B}^{\mathcal{S}}$; yet the notation $2^{\mathcal{S}}$ will be used to make it explicit that the powerset is referred to.

Observation i: If and only if there exists $v \in \mathcal{S} \cap \mathcal{S}'$ with $\psi_v \neq \psi'_v$ then $\psi\psi' \neq \psi'\psi$.

Observation ii: If $\mathcal{S} \subseteq \mathcal{S}'$ then $\psi\psi' = \psi'$;

Observation iii: Re-colouring is associative: $(\psi\psi')\psi'' = \psi(\psi'\psi'')$.

Following the notation $\chi^{\mathcal{S}}$, the effect of $\psi\chi^{\mathcal{S}}$ is that all elements of $\mathcal{S}$ are re-coloured with χ. If $\mathcal{S}$ contains only one element v, then this may be further abbreviated to $\psi\chi^v$.

Lemma 2.2.2. Let $\psi \in X^{\mathcal{S}}$ and $\psi' \in X^{\mathcal{S}'}$. Combining re-colouring with projections, we have:

i. $(\psi \searrow \mathcal{S}'')\psi' = (\psi\psi') \searrow (\mathcal{D}(\psi') \cup \mathcal{S}'')$;

ii. if $\mathcal{D}(\psi') \subseteq \mathcal{S}''$ then $(\psi \searrow \mathcal{S}'')\psi' = (\psi\psi') \searrow \mathcal{S}''$;

iii. If $\mathcal{D}(\psi') \cap \mathcal{S}'' = \varnothing$ then $(\psi\psi') \searrow \mathcal{S}'' = \psi \searrow \mathcal{S}''$;

iv. $\psi \searrow \mathcal{S}_0 \searrow \mathcal{S}_1 \searrow \ldots \searrow \mathcal{S}_{k-1} = \phi^{\mathcal{S}_{k-1}}(\psi \searrow \mathcal{S}^*) = (\psi \searrow \mathcal{S}_{k-1})\phi^{\mathcal{S}_{k-1} \setminus \mathcal{S}^*}$ where $\mathcal{S}^* = \bigcap_{i \in \mathbb{Z}_k} \mathcal{S}_i$;

v. $\psi \searrow \mathcal{S}'' \searrow \mathcal{D}(\psi) = \psi\phi^{\mathcal{D}(\psi) \setminus \mathcal{S}''}$.

Proof. Assertions i, iii, and iv can be verified by checking all the cases based on membership of the sets and domains. Assertions ii and v are special cases of i and iv, respectively. □

Definition 2.2.3. Let $\psi, \psi' \in X^{\mathcal{S}}$, then ψ' is **a child** of ψ if ψ' can be obtained by assigning a colour to an uncoloured element in ψ. In such a case, ψ is **a parent** of ψ'. This is denoted as $\psi \rightarrow \psi'$ and $\psi' \leftarrow \psi$.

$$\psi \rightarrow \psi' \overset{\text{def}}{\Longleftrightarrow} \exists_{v \in \psi^{-1}(\phi), \chi \in \check{X}} \left[\psi' = \psi\chi^v\right]$$

Definition 2.2.4. Let $\psi, \psi' \in X^{\mathcal{S}}$. If there is a sequence $\psi \rightarrow \cdots \rightarrow \psi'$, then ψ' is **a descendant** of ψ and ψ is **an ancestor of** ψ'. This is denoted $\psi' \succ \psi$ and $\psi \prec \psi'$. The notations $\psi' \succcurlyeq \psi$ and $\psi \preccurlyeq \psi'$ refer to $\psi' \succ \psi \vee \psi' = \psi$.

$$\psi' \succcurlyeq \psi \overset{\text{def}}{\Longleftrightarrow} \forall_{v \in \mathcal{S}}\left[\psi_v = \phi \vee \psi'_v = \psi_v\right] \Longleftrightarrow \exists_{\mathcal{S}' \subseteq \mathcal{S}}\left[\psi = \psi'\phi^{\mathcal{S}'}\right]$$

If $\psi' \succcurlyeq \psi$ and ψ' is a complete colouring, then ψ' is **a completion** of ψ. This is denoted $\psi' \trianglerighteq \psi$ and $\psi \trianglelefteq \psi'$.

2.3 Functions

Within the sets $\mathbb{B}$ and $\mathbb{T}$, the symbols $\wedge$, $\vee$, and $\equiv$ are used for conjunction, disjunction, and equivalence, respectively. Refer to Table 2.3.1 for their values; as can be seen, the functions can also be computed by taking respectively the minimum, maximum, and product of the arguments. We therefore have the following trivial lemma:

Lemma 2.3.1. Let $\psi, \psi' \in \mathbb{B}^{\mathcal{S}}$, then $\psi = \psi'$ if and only if $\bigwedge_{v \in \mathcal{S}}(\psi_v \cdot \psi'_v) = \text{TRUE}$.

t	numerical	$-t$
FALSE	-1	TRUE
ϕ	0	ϕ
TRUE	$+1$	FALSE

$\wedge$	FALSE	ϕ	TRUE
FALSE	FALSE	FALSE	FALSE
ϕ	FALSE	ϕ	ϕ
TRUE	FALSE	ϕ	TRUE

$\vee$	FALSE	ϕ	TRUE
FALSE	FALSE	ϕ	TRUE
ϕ	ϕ	ϕ	TRUE
TRUE	TRUE	TRUE	TRUE

$\equiv$	FALSE	ϕ	TRUE
FALSE	TRUE	ϕ	FALSE
ϕ	ϕ	ϕ	ϕ
TRUE	FALSE	ϕ	TRUE

Table 2.3.1: Boolean functions extended to $\mathbb{T}$.

A **disjunctive clause** is a formula of the form

$$\mathcal{C}(\psi) = \big(\psi(v_0) \equiv \chi_0)\big) \vee \big(\psi(v_1) \equiv \chi_1)\big) \vee \ldots \vee \big(\psi(v_{k-1}) \equiv \chi_{k-1}\big)$$

where $\psi(v) \equiv \chi$ is TRUE if $\psi_v = \chi$ and FALSE if $\psi_v \neq \chi$. A formula is given in **conjunctive normal form** (CNF) if it is a conjunction of disjunctive clauses. The notions of **conjunctive clause** and **disjunctive normal form** (DNF) are defined analogously. If the function's domain is Boolean, namely $\mathbb{B}^{\mathcal{S}}$ for some set $\mathcal{S}$, then the definitions correspond to the standard definitions of CNF and DNF, where χ_i = TRUE and χ_j = FALSE correspond to literals occurring in positive form and in negative form, respectively.

Definition 2.3.2. Let $\mathcal{C}$ be a clause occurring in a CNF formula. Then $\mathcal{C}$ is **reducible** if a clause $\mathcal{C}'$ appears in the CNF such that $\mathcal{C}$ is of the form $(\mathcal{C}' \wedge \ldots)$. Similarly, if $\mathcal{C}$ is a clause in a DNF formula, then $\mathcal{C}$ is reducible if the DNF contains a clause $\mathcal{C}'$ such that $\mathcal{C} = (\mathcal{C}' \vee \ldots)$.

> *Observation i:* If the CNF or the DNF of a Boolean function contains a reducible clause $\mathcal{C}$, then $\mathcal{C}$ may be omitted, since according to the Boolean absorption law we have $t_1 \wedge (t_1 \vee t_2) = t_1 \vee (t_1 \wedge t_2) = t_1$.
>
> *Observation ii:* If a CNF clause $\mathcal{C}$ contains a literal that is equal to FALSE, then the literal may be omitted. If $\mathcal{C}$ contains a literal that is equal to TRUE, then the entire clause may be omitted. If $\mathcal{C}$ contains only literals equal to FALSE, then the formula is always equal to FALSE. For a DNF clause the same assertions hold with the roles of TRUE and FALSE interchanged.

When there is some element of $\mathcal{S}$ that has no influence on the value of f, then this element is dead.

Definition 2.3.3. Let $f : X^{\mathcal{S}} \to \mathbb{B}$ and $v \in \mathcal{S}$. Then v is **dead** in f if for all $\psi \in X^{\mathcal{S}}$ and all $\chi \in X$ we have $f(\psi) = f(\psi\chi^v)$. If v is not dead in f then it is **live** in f.

Informally, an element being dead with respect to a function means that the element's colour cannot influence the value of the function.

Throughout the remainder of the text, functions will often be applied to arguments that do not appear in the domain of the function. By default, it is then assumed that the argument is first projected onto the domain of the function.

Definition 2.3.4. Let $\psi \in X^{\mathcal{S}}$ and $f : X^{\mathcal{S}'} \to \mathbb{B}$. Then $f(\psi) \stackrel{\text{def}}{=\!=} f(\psi \searrow \mathcal{S}')$.

2.4 Isotone Functions

When some complete ordering is chosen for X, the concepts of monotone elements and functions arise.

Definition 2.4.1. Let $f : X^{\mathcal{S}} \to \mathbb{B}$. If X is an ordered set, then f is **increasing** in $v \in \mathcal{S}$ if

$$\forall_{\chi_+,\chi_- \in X} \forall_{\psi \in X^{\mathcal{S}}} \left[\chi_+ \geqslant \chi_- \implies f(\psi\chi_+^v) \geqslant f(\psi\chi_-^v)\right]. \tag{2.4.1}$$

Similarly, f is **decreasing** in v if $\chi_+ \geqslant \chi_- \implies f(\psi\chi_+^v) \leqslant f(\psi\chi_-^v)$ for all $\psi \in X^{\mathcal{S}}$. If f is increasing or decreasing in v then f is **monotone in** v.

Observation i: If $X = \mathbb{B}$ then f is increasing in v if and only if for all $\psi \in \mathbb{B}^{\mathcal{S}}$ we have $f(\psi\mathrm{F}^v) \implies f(\psi\mathrm{T}^v)$.

Observation ii: If and only if f is both increasing and decreasing in v then v is dead in f.

Definition 2.4.2. A function $f : X^{\mathcal{S}} \to \mathbb{B}$ is monotone if it is monotone in all elements of $\mathcal{S}$, and f is **isotone** if it is increasing in all elements of $\mathcal{S}$.

Observation i: Function f is isotone if and only if $\forall_{\psi,\psi' \in X^{\mathcal{S}}} [\psi \geqslant \psi' \implies f(\psi) \geqslant f(\psi')]$.

Observation ii: Any monotone function can be made isotone by replacing all decreasing elements by their negations. So without loss of generality all monotone Boolean functions can be considered isotone.

Observation iii: If a Boolean function $f : \mathbb{B}^{\mathcal{S}} \to \mathbb{B}$ is isotone, then an element $v \in \mathcal{S}$ is live if and only if there is a $\psi \in \mathbb{B}^{\mathcal{S}}$ such that $f(\psi\mathrm{T}^v) = +1$ and $f(\psi\mathrm{F}^v) = -1$.

Definition 2.4.3. Let $\mathcal{F}$ be a family of subsets of $\mathcal{S}$, and let X be a set of colours with $X' \subseteq X$. The **coalition function** $\textsc{coal}(\mathcal{F}; X') : X^{\mathcal{S}} \to \mathbb{B}$ is defined as:

$$\textsc{coal}(\mathcal{F}; X') : \xi \mapsto \exists_{\mathcal{S}' \in \mathcal{F}} \forall_{v \in \mathcal{S}'} \left[\xi(v) \in X'\right].$$

In this function, the members of $\mathcal{F}$ are called **coalitions**.

Observation i: The function $\xi \mapsto \forall_{\mathcal{S} \in \mathcal{F}} \exists_{v \in \mathcal{S}} \left[\xi(v) \in X'\right]$ is equal to $-\textsc{coal}(\mathcal{F}; X \setminus X')$.

The coalition function thus reports whether the family of subsets contains a particular subset that is coloured entirely with colours from X'. This is asking a property of all of the elements of some of the subsets, which is equivalent to asking the opposite property of some of the elements from all of the subsets; namely, if the coalition function is false, then all of the subsets contain some elements that are coloured with colours from $X \setminus X'$.

Theorem 2.4.4. Let $f : \mathbb{B}^{\mathcal{S}} \to \mathbb{B}$. The following statements are equivalent:

i. f is isotone;

ii. there exists a family $\mathcal{F}$ of subsets of $\mathcal{S}$ such that $f(\psi)$ is TRUE for $\chi \in X^{\mathcal{S}}$ if and only if there is a $\mathcal{S}' \in \mathcal{F}$ with $\psi_v = \text{TRUE}$ for every $v \in \mathcal{S}'$;

iii. there exists a family $\mathcal{F}$ of subsets of $\mathcal{S}$ such that $f(\psi)$ is TRUE for $\chi \in X^{\mathcal{S}}$ if and only if for every subset $\mathcal{S}' \in \mathcal{F}$ there is a $v \in \mathcal{S}'$ with $\psi_v = \text{TRUE}$;

iv. f has a DNF representation in which only positive literals occur;

v. f has a CNF representation in which only positive literals occur.

Proof. The trajectory is as follows: v $\Longrightarrow$ iii $\Longrightarrow$ i $\Longrightarrow$ ii $\Longrightarrow$ iv $\Longrightarrow$ v.

v $\Longrightarrow$ iii: Let the CNF representation be $\bigwedge_{i \in \mathbb{Z}_k} \mathcal{C}_i$ where each $\mathcal{C}_i$ is a disjunctive clause. Put $\mathcal{F} = \{\mathcal{S}_i^*\}_{i \in \mathbb{Z}_k}$ where $\mathcal{S}_i^* \subseteq \mathcal{S}$ contains the element indices occurring in $\mathcal{C}_i$. Since all elements occur in $\mathcal{C}_i$ only in positive form, $\mathcal{F}$ meets the requirements of statement iii.

iii $\Longrightarrow$ i: Let $\psi \in \mathbb{B}^{\mathcal{S}}$ and $v \in \mathcal{S}$. If $f(\psi\text{F}^v) = \text{TRUE}$ then for each $\mathcal{S}^* \in \mathcal{F}$ there is a $w \in \mathcal{S}^*$ such that $\psi\text{F}^v(w) = \text{TRUE}$, and so $\psi\text{T}^v(w) = \text{TRUE}$. Therefore $f(\psi\text{T}^v) = \text{TRUE}$.

i $\Longrightarrow$ ii: Let f be isotone, and let $\psi^{(0)}, \psi^{(1)}, \ldots, \psi^{(k-1)} \in X^{\mathcal{S}}$ be a list of all colourings whose f-value is $+1$. This list contains at most $2^{|\mathcal{S}|}$ elements. Define $\mathcal{F} = \{\mathcal{S}_i^*\}_{i \in \mathbb{Z}_k}$ where each $\mathcal{S}_i^* = {\psi^{(i)}}^{-1}(\text{TRUE})$. Then $f = \text{COAL}(\mathcal{F}; \{\text{TRUE}\})$.

ii $\Longrightarrow$ iv: Let $f = \text{COAL}(\mathcal{F}; \{\text{TRUE}\})$ with $\mathcal{F} = \{\mathcal{S}_i^*\}_{i \in \mathbb{Z}_k}$. Define $\mathcal{C}_i$ as the disjunction containing the positive forms of precisely those elements whose indices occur in $\mathcal{S}_i^*$. Then $\bigvee_{i \in \mathbb{Z}_k} \mathcal{C}_i$ is a DNF representation of f.

iv $\Longrightarrow$ v: It is well known that any DNF can be transformed into a CNF in which exactly the same literals occur. □

2.5 Graphs

When $\mathcal{G}$ is a graph, the **vertex set** and **edge set** of $\mathcal{G}$ are denoted as $\mathcal{V}(\mathcal{G})$ and $\mathcal{E}(\mathcal{G})$. Unless stated otherwise, all graphs shall be simple undirected graphs with no loops or multiple edges. The number of vertices of the graph is denoted as $|\mathcal{G}| \stackrel{\text{def}}{=\!=} |\mathcal{V}(\mathcal{G})|$. The **complete graph** on $\mathcal{S}$ is the graph $\mathcal{K}_{\mathcal{S}}$ whose vertex set is $\mathcal{S}$ and whose edge set contains all edges. Given two graphs $\mathcal{G}$ and $\mathcal{G}'$, write $\mathcal{G}' \subseteq \mathcal{G}$ when $\mathcal{V}(\mathcal{G}') \subseteq \mathcal{V}(\mathcal{G})$ and $\mathcal{E}(\mathcal{G}') \subseteq \mathcal{E}(\mathcal{G})$.

Vertices $v, w \in \mathcal{V}(\mathcal{G})$ are **adjacent** in $\mathcal{G}$ if $(v, w) \in \mathcal{E}(\mathcal{G})$. This is denoted as $v \overset{\mathcal{G}}{\sim} w$. A **clique** in a graph is a set of vertices such that each pair of vertices is adjacent. The **neighbourhood** of v is the set $\mathcal{N}_{\mathcal{G}}(v) \overset{\text{def}}{=\!=} \{w \in \mathcal{G} : v \overset{\mathcal{G}}{\sim} w\}$. If no ambiguity as to $\mathcal{G}$ is possible, the notations $v \sim w$ and $\mathcal{N}(v)$ may be used. For a set $\mathcal{S} \subseteq \mathcal{V}(\mathcal{G})$, the neighbourhood of $\mathcal{S}$ is the set $\mathcal{N}(\mathcal{S}) \overset{\text{def}}{=\!=} \{w \notin \mathcal{S} : \exists_{v \in \mathcal{S}}[v \sim w]\} = \left(\bigcup_{v \in \mathcal{S}} \mathcal{N}(v)\right) \setminus \mathcal{S}$. A vertex or a set of vertices is **simplicial** if its neighbourhood is a clique.

For $\mathcal{S} \subseteq \mathcal{V}(\mathcal{G})$, the **induced subgraph** $\mathcal{G}(\mathcal{S})$ is defined as the graph whose vertex set is $\mathcal{S}$ and whose edge set contains all edges in $\mathcal{G}$ that contain two vertices in $\mathcal{S}$. The **graph union** of two graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ is the graph $\mathcal{G}_1 \cup \mathcal{G}_2$ whose vertex set is $\mathcal{V}(\mathcal{G}_1) \cup \mathcal{V}(\mathcal{G}_2)$ and whose edge set is $\mathcal{E}(\mathcal{G}_1) \cup \mathcal{E}(\mathcal{G}_2)$.

A **path** is a sequence of vertices $\mathcal{P} = (v_0, v_1, \ldots, v_{k-1})$ such that $v_0 \sim v_1 \sim \cdots \sim v_{k-1}$. Two vertices $v, w \in \mathcal{G}$ are **connected** if there exists a path $(v, \ldots, w)$; such a path is called a v-w path. A set $\mathcal{S} \subseteq \mathcal{V}(\mathcal{G})$ is a v-w **connector** if it contains a v-w path, and $\mathcal{S}$ is a v-w **separator** if every v-w path intersects $\mathcal{S}$. A connector or separator is **minimal** if no subset has the same property.

Definition 2.5.1. The operation of **deleting** a set $\mathcal{S}$ of vertices from a graph $\mathcal{G}$ consists of removing $\mathcal{S}$ from $\mathcal{V}(\mathcal{G})$ and any corresponding edge from $\mathcal{E}(\mathcal{G})$. This is denoted as $\mathcal{G}\backslash\mathcal{S}$, so that $\mathcal{G}\backslash\mathcal{S} \overset{\text{def}}{=\!=} \mathcal{G}(\mathcal{V}(\mathcal{G}) \setminus \mathcal{S})$. The operation of **contracting** $\mathcal{S}$ consists of first adding edges between all pairs of vertices in the neighbourhood of $\mathcal{S}$, and then deleting $\mathcal{S}$. This is denoted as $\mathcal{G}/\mathcal{S}$, so that $\mathcal{G}/\mathcal{S} \overset{\text{def}}{=\!=} (\mathcal{G} \cup \mathcal{K}_{\mathcal{N}(\mathcal{S})})\backslash\mathcal{S}$.

Observation i: $(\mathcal{G}\backslash\mathcal{S}_1)\backslash\mathcal{S}_2 = \mathcal{G}\backslash(\mathcal{S}_1 \cup \mathcal{S}_2) = (\mathcal{G}\backslash\mathcal{S}_2)\backslash\mathcal{S}_1$.

Observation ii: $(\mathcal{G}/\mathcal{S}_1)/\mathcal{S}_2 = \mathcal{G}/(\mathcal{S}_1 \cup \mathcal{S}_2) = (\mathcal{G}/\mathcal{S}_2)/\mathcal{S}_1$.

Observation iii: $(\mathcal{G}\backslash\mathcal{S}_1)/\mathcal{S}_2 = (\mathcal{G}/\mathcal{S}_2)\backslash\mathcal{S}_1$.

2.6 Hex Notation

The regular Hex board of size $n \times n$ shall be denoted as $\mathbb{X}_n$.[4] The elements on a Hex board are also called **cells**. A connected set of cells all of which have the same colour will be called **a chain**.

Cells on a Hex board are designated with chess-like row and column names as in the example shown in Figure 2.1. The moves played in the sample game on the right in Figure 2.1 are listed in Table 2.6.1.

For algorithmic implementation considerations it is useful to recognize that the goals for *both* players on $\mathbb{X}_5$ are equivalent to connecting any pair of opposite corner cells on the $\mathbb{X}_7$ diagram in Figure 2.2. When representing a Hex board this way, the black and white stones on the outer rows and columns are called the **border pieces**.

[4] This notation is chosen to avoid confusion with the symbol $\mathbb{H}$ commonly used for quaternions, and to line up with the symbol $\mathbb{Y}$, to be used for Y boards (see Section 4.6).

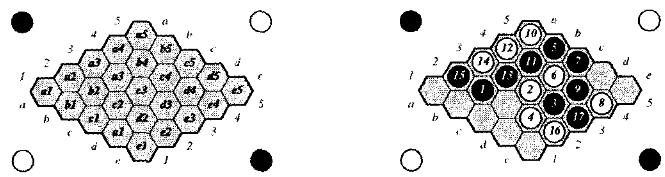


Figure 2.1: Coordinate system for $\mathbb{X}_5$ (left), and a sample game (right).

Black		White	
1.	b2	2.	c3
3.	d3	4.	d2
5.	b5	6.	c4
7.	c5	8.	e4
9.	d4	10.	a5
11.	b4	12.	a4
13.	b3	14.	a3
15.	a2	16.	e2
17.	e3		

Table 2.6.1: Moves played in the game shown in Figure 2.1.

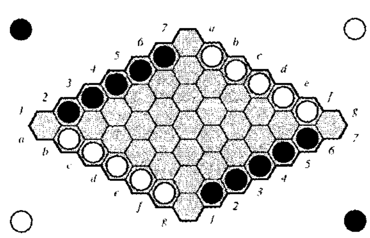


Figure 2.2: Border pieces on $\mathbb{X}_7$, making this position equivalent to the empty board $\mathbb{X}_5$.

Chapter 3

Set Colouring Games

Set colouring games form a large class of abstract games, containing Hex as a special case. The class is a slight generalization of the class of games played on Boolean formulas. Games with Boolean *coefficients* were studied by Ostmann [75, 76, 77], who was mostly concerned with games whose goal was to reach a weighted majority. Games where the goal is itself an arbitrary Boolean function of the coefficients were specifically introduced by Zhao and Müller under the name GAME-SAT [107].[1]

3.1 Game Definitions

Within the context of set colouring games, the existence of two fixed sets is assumed: X, containing the colours, and $\mathfrak{C}$, containing the players.

Definition 3.1.1. The set of **colours** is X, which contains ϕ and at least one pure colour: $X \neq \check{X} \neq \varnothing$. If $\check{X}$ is ordered, then F and T denote the minimum and maximum pure colour.

Definition 3.1.2. The set of **players** is $\mathfrak{C} \overset{\text{def}}{=\!=} \{\text{MIN}, \text{MAX}\}$. For $\mathfrak{c} \in \mathfrak{C}$ the notation $\bar{\mathfrak{c}}$ refers to $\mathfrak{c}$'s opponent, so that $\mathfrak{C} = \{\mathfrak{c}, \bar{\mathfrak{c}}\}$. Define $\lambda : \mathfrak{C} \to \mathbb{B}$ by putting $\lambda(\text{MAX}) = +1$ and $\lambda(\text{MIN}) = -1$, and for $t \in \mathbb{B}$ define $t \cdot \mathfrak{c} \overset{\text{def}}{=\!=} \lambda^{-1}\big(t \cdot \lambda(\mathfrak{c})\big)$.

> *Observation i:* For any $\mathfrak{c} \in \mathfrak{C}$ we have $\lambda(\bar{\mathfrak{c}}) = -\lambda(\mathfrak{c})$.
>
> *Observation ii:* For any $\mathfrak{c} \in \mathfrak{C}$ we have $(\lambda(\mathfrak{c}))^2 = 1$.

Definition 3.1.3. A set colouring game consists of a colour space $X^{\mathcal{S}}$ where $\mathcal{S}$ is finite, and a function $f : \check{X}^{\mathcal{S}} \to \mathbb{B}$. This set colouring game will be denoted as $\langle X^{\mathcal{S}}, f \rangle$. The function f is its **scoring function**, and $\mathcal{S}$ is the set of **elements**.

[1] See Section 4.2.

The game starts with all elements uncoloured. There is no fixed convention as to which player moves first. The two players take turns colouring a previously uncoloured element of $\mathcal{S}$ with a pure colour from X. Players may never uncolour an element. This ensures that the game ends after exactly $|\mathcal{S}|$ moves with a complete colouring. The function f then indicates the **outcome** of the game. Player MAX tries to maximize the outcome by making it equal to TRUE, while player MIN has the opposite goal. The number $|\mathcal{S}|$ is the game's **dimension**, and the game may be called **even** or **odd** according to the parity of its dimension.

In most games there will only be two pure colours, in which case we can set $X = \mathbb{T}$. Any GAME-SAT instance is such a game. The definitions and theorems in this chapter are equally valid for games that use more than two pure colours, so they will be kept general. The colourings that occur in the game $\langle X^{\mathcal{S}}, f\rangle$ are all in the domain $X^{\mathcal{S}}$; the game starts with the colouring $\phi^{\mathcal{S}}$ and ends with a colouring in $\check{X}^{\mathcal{S}}$.

Definition 3.1.4. Let $\Gamma = \langle X^{\mathcal{S}}, f\rangle$. If $f : \xi \mapsto t$ for $t \in \mathbb{B}$, then Γ is **trivial**, and is also denoted as $\langle X^{\mathcal{S}}, t\rangle$. The game $\langle \varnothing, t\rangle$ is defined as $\langle X^{\varnothing}, t\rangle$.

So in particular we have the game $\langle X^{\mathcal{S}}, +\rangle$ whose outcome is $+1$ no matter what moves were played, and similarly the outcome of $\langle X^{\mathcal{S}}, -\rangle$ is always -1. The "game" $\langle \varnothing, t\rangle$ ends after zero moves with the outcome t.

In some cases it will be useful to consider the effects of adding a dead element to the game. This will be called a *starred game*.

Definition 3.1.5. Let $\Gamma = \langle X^{\mathcal{S}}, f\rangle$. The game $\Gamma*$ is defined as $\langle X^{\mathcal{S}^*}, f\rangle$, with $\mathcal{S}^* = \mathcal{S} + w$ for some arbitrary $w \notin \mathcal{S}$. The game $\Gamma*$ is the **starred variant** of Γ, and w is the **added element**.

Note that the starred game actually uses the function $f \circ \text{PROJ}_{\mathcal{S}^* \to \mathcal{S}}$, as per Definition 2.3.4.

As will be seen later, the starred version of a game may have a different outcome, even though the added element is dead. The notation can be iterated to define $\Gamma ** \stackrel{\text{def}}{=\!=} (\Gamma*)*$, though fortunately it will turn out that there will be no need to do so.

3.2 Moves and Positions

According to standard game terminology, *a position* specifies all the necessary information about the status of the game, and a *move* leads from one position to the next. Set colouring games are **perfect information games**, meaning no relevant information is hidden from any of the players.

Definition 3.2.1. For $\psi \in X^{\mathcal{S}}$, the sets $\mathcal{U}(\psi)$ and $\mathcal{A}(\psi)$ are the sets of uncoloured and coloured elements, respectively.

$$\mathcal{U}(\psi) \stackrel{\text{def}}{=\!=} \psi^{-1}(\phi),$$
$$\mathcal{A}(\psi) \stackrel{\text{def}}{=\!=} \mathcal{D}(\psi) \setminus \mathcal{U}(\psi).$$

The **purified** version of ψ is $\overline{\psi} \stackrel{\text{def}}{=\!=} \psi \searrow \mathcal{A}(\psi)$.

Observation i: $\mathcal{U}(\psi\psi') = \mathcal{U}(\psi') \cup \big(\mathcal{U}(\psi) \setminus \mathcal{A}(\psi')\big)$, and therefore $\mathcal{U}(\psi\psi') = \mathcal{U}(\psi)$ if and only if $\mathcal{U}(\psi') \subseteq \mathcal{U}(\psi)$ and $\mathcal{A}(\psi') \cap \mathcal{U}(\psi) = \varnothing$;

Observation ii: Observation i also holds with $\mathcal{U}$s and $\mathcal{A}$s interchanged;

Observation iii: For $v \in \mathcal{S}$ and $\chi \in X$:

$$|\mathcal{U}(\psi\chi^v)| = \begin{cases} |\mathcal{U}(\psi)| + 1 & \text{if} \quad \psi_v \neq \phi = \chi; \\ |\mathcal{U}(\psi)| - 1 & \text{if} \quad \psi_v = \phi \neq \chi; \\ |\mathcal{U}(\psi)| & \text{if} \quad \psi_{gel} \neq \phi \neq \chi. \end{cases}$$

Observation iv: $\mathcal{U}(\psi \searrow \mathcal{S}') = \big(\mathcal{U}(\psi) \cap \mathcal{S}'\big) \cup \big(\mathcal{S}' \setminus \mathcal{S}\big)$.

Observation v: $\mathcal{A}(\psi \searrow \mathcal{S}') = \mathcal{A}(\psi) \cap \mathcal{S}'$.

Observation vi: $\psi \in \check{X}^{\mathcal{S}} \iff \mathcal{U}(\psi) = \varnothing \iff \mathcal{A}(\psi) = \mathcal{S}$.

Definition 3.2.2. For $\psi \in X^{\mathcal{S}}$, **a legal move** in ψ is a colouring that assigns a pure colour to exactly one uncoloured element in ψ. The set of all legal moves in ψ is $\mathfrak{M}(\psi)$.

$$\mathfrak{M}(\psi) \stackrel{\text{def}}{=\!=} \bigcup_{v \in \mathcal{U}(\psi)} \check{X}^v. \tag{3.2.1}$$

For $\mathfrak{m} = \chi^v \in \mathfrak{M}(\psi)$, write $\psi\mathfrak{m} \stackrel{\text{def}}{=\!=} \psi\chi^v$.

Observation i: $\mathfrak{M}(\psi) \subseteq \mathfrak{M}(\phi^{\mathcal{S}})$;

Observation ii: $\mathfrak{M}(\psi\psi') = \mathfrak{M}(\psi)$ if and only if $\mathcal{A}(\psi\psi') = \mathcal{A}(\psi)$, also see Observation 3.2.1:i;

Observation iii: if $\mathfrak{m} \in \mathfrak{M}(\psi)$ then $\psi \to \psi\mathfrak{m}$ and $|\mathcal{U}(\psi\mathfrak{m})| = |\mathcal{U}(\psi)| - 1$.

Definition 3.2.3. A position on a set $\mathcal{S}$ is a pair $\mathfrak{p} = (\psi, \mathfrak{c})$ where $\psi \in X^{\mathcal{S}}$ and $\mathfrak{c} \in \mathfrak{C}$. The component $\mathfrak{c}$ indicates which player is to move next. The set of all positions on $\mathcal{S}$ is denoted as $\mathfrak{P}_{\mathcal{S}}$.

$$\mathfrak{P}_{\mathcal{S}} \stackrel{\text{def}}{=\!=} X^{\mathcal{S}} \times \mathfrak{C}. \tag{3.2.2}$$

If $\psi = \phi^{\mathcal{S}}$ then $\mathfrak{p}$ is **an initial position**. If ψ is a complete colouring then $\mathfrak{p}$ is **a final position**. The following shorthand notations are used:

- $\mathcal{U}(\mathfrak{p}) \stackrel{\text{def}}{=\!=} \mathcal{U}(\psi)$;

- $\mathfrak{M}(\mathfrak{p}) \stackrel{\text{def}}{=} \mathfrak{M}(\psi)$.[2]
- $\mathfrak{p} \searrow \mathcal{S}' \stackrel{\text{def}}{=} (\psi \searrow \mathcal{S}', \mathfrak{c})$.
- $\mathfrak{p}\psi' \stackrel{\text{def}}{=} (\psi\psi', \mathfrak{c}) \in \mathfrak{P}_{\mathcal{S}}$;
- for $\mathfrak{p} = (\psi, \mathfrak{c})$ and $t \in \mathbb{B}$ write $t\mathfrak{p} = (\psi, t\mathfrak{c})$, so that in particular $-\mathfrak{p} = (\psi, \bar{\mathfrak{c}})$;
- if $\mathfrak{p} = (\psi, \mathfrak{c})$ is a final position then $f(\mathfrak{p}) \stackrel{\text{def}}{=} f(\psi)$.

Definition 3.2.4. For $\psi \in X^{\mathcal{S}}$, $\mathfrak{p} = (\psi, \mathfrak{c}) \in \mathfrak{P}_{\mathcal{S}}$, and $\mathfrak{m} \in \mathfrak{M}(\mathfrak{p})$, define $\mathfrak{p} \oplus \mathfrak{m} \stackrel{\text{def}}{=} (\psi\mathfrak{m}, \bar{\mathfrak{c}})$. The definitions of parent and child for colourings[3] apply similarly to positions; $\mathfrak{p} \to \mathfrak{p}'$ and $\mathfrak{p}' \leftarrow \mathfrak{p}$ if and only if $\mathfrak{p}' = \mathfrak{p} \oplus \mathfrak{m}$ for some $\mathfrak{m} \in \mathfrak{M}(\mathfrak{p})$.

Observation i: If $\mathfrak{m} = \chi^v \in \mathfrak{M}(\mathfrak{p})$ and $\psi' \in X^{\mathcal{S}-v}$ then $\mathfrak{p}\psi' \oplus \mathfrak{m} = (\mathfrak{p} \oplus \mathfrak{m})\psi'$.

Definition 3.2.4 of course implies the convention that players take turns colouring elements. Note the difference between $\mathfrak{p}\psi'$ and $\mathfrak{p} \oplus \mathfrak{m}$: the former is defined for any colouring ψ', whereas the latter is only defined for colourings that represent legal moves and also includes a switch from $\mathfrak{c}$ to $\bar{\mathfrak{c}}$.

Definition 3.2.5. A transition is a function $\mathfrak{t} : \mathfrak{P}_{\mathcal{S}} \to \mathfrak{P}_{\mathcal{S}}$ such that for every $\mathfrak{p} \in \mathfrak{P}_{\mathcal{S}}$ we have $\mathfrak{p} \to \mathfrak{t}(\mathfrak{p})$ if $\mathfrak{p}$ is not a final position, and $\mathfrak{t}(\mathfrak{p}) = \mathfrak{p}$ otherwise. The value $f(\mathfrak{p}, \mathfrak{t})$ is defined as $f(\mathfrak{t}^k(\mathfrak{p}))$ with $k = |\mathcal{U}(\mathfrak{p})|$.

Observation i: $|\mathcal{U}(\mathfrak{t}(\mathfrak{p}))| = |\mathcal{U}(\mathfrak{p})| - 1$ if $\mathfrak{p}$ is not a final position.

Observation ii: By induction from Observation 3.2.5:i, $\mathfrak{t}^k(\mathfrak{p})$ is a final position, so $f(\mathfrak{p}, \mathfrak{t})$ is well defined.

Observation iii: If $\mathfrak{p} = (\psi, \mathfrak{c})$ is itself a final position then $f(\mathfrak{p}, \mathfrak{t}) = f(\psi)$.

3.3 Strategies

The expression *strategy* is often used in an informal fashion in game play, vaguely referring to some more or less well defined plan that one of the players may have in mind for selecting moves. Yet the concept can be defined precisely.

Definition 3.3.1. A **strategy** is a function $\mathfrak{s} : X^{\mathcal{S}} \to \mathfrak{M}(\phi^{\mathcal{S}})$ such that for every *incomplete* $\psi \in X^{\mathcal{S}}$ the value $\mathfrak{s}(\psi)$ is a legal move in ψ. The set of all strategies on $\mathcal{S}$ is denoted as $\mathfrak{S}_{\mathcal{S}}$.

Note that the definition of a strategy does not specify anything about optimal play; it is simply a function that returns a legal move whenever asked to do so. It has no notion of which player is to move next, its domain being $X^{\mathcal{S}}$ rather than $\mathfrak{P}_{\mathcal{S}}$.

---

[2]This makes GAME-SAT *almost* an "impartial game" – see Section 7.1.
[3]See Definition 2.2.3.

Definition 3.3.2. Let $\mathfrak{s} \in \mathfrak{S}_{\mathcal{S}}$. The **strategy transition** associated with $\mathfrak{s}$ is the transition $\mathfrak{t} : \xi \mapsto \xi \oplus \mathfrak{s}(\psi) : \mathfrak{P}_{\mathcal{S}} \to \mathfrak{P}_{\mathcal{S}}$.

A transition results when there is only one player playing the game, using the specified strategy. In a two player game the two players would typically not choose to play the same move in a given colouring, so they will use different strategies. A concept is needed for the transition that occurs when both players use their own strategy.

Definition 3.3.3. Let $\mathfrak{s}_{\text{MIN}}, \mathfrak{s}_{\text{MAX}} \in \mathfrak{S}_{\mathcal{S}}$ be two strategies. The transition $\mathfrak{t} = \mathfrak{s}_{\text{MIN}} * \mathfrak{s}_{\text{MAX}} : \mathfrak{P}_{\mathcal{S}} \to \mathfrak{P}_{\mathcal{S}}$ is defined as:

$$\mathfrak{t}(\psi, \mathfrak{c}) \stackrel{\text{def}}{=\!=} (\psi, \mathfrak{c}) \oplus \begin{cases} \mathfrak{s}_{\text{MIN}}(\psi) & \text{if } \mathfrak{c} = \text{MIN}; \\ \mathfrak{s}_{\text{MAX}}(\psi) & \text{if } \mathfrak{c} = \text{MAX}. \end{cases}$$

Definition 3.3.4. Let $\Gamma = \langle X^{\mathcal{S}}, f \rangle$ and $\mathfrak{p} \in \mathfrak{P}_{\mathcal{S}}$. If $\mathfrak{s}_{\text{MIN}} \in \mathfrak{S}_{\mathcal{S}}$ has the property

$$\forall_{\mathfrak{s} \in \mathfrak{S}_{\mathcal{S}}} \Big[f(\mathfrak{p}, \mathfrak{s}_{\text{MIN}} * \mathfrak{s}) = -1 \Big]$$

then $\mathfrak{s}_{\text{MIN}}$ is an Γ-**winning strategy** for MIN in $\mathfrak{p}$. If $\mathfrak{s}_{\text{MAX}} \in \mathfrak{S}_{\mathcal{S}}$ has the property

$$\forall_{\mathfrak{s} \in \mathfrak{S}_{\mathcal{S}}} \Big[f(\mathfrak{p}, \mathfrak{s} * \mathfrak{s}_{\text{MAX}}) = +1 \Big]$$

then $\mathfrak{s}_{\text{MAX}}$ is an Γ-winning strategy for MAX in $\mathfrak{p}$.

Observation i: In any position at most one of the two players can have a Γ-winning strategy.

3.4 Homomorphisms

The following definitions follow those given by Yamasaki for a different class of games[4] [102].

Definition 3.4.1. Let $\Gamma = \langle X^{\mathcal{S}}, f \rangle$ and $\Gamma' = \langle X^{\mathcal{S}'}, f' \rangle$. A function $h : X^{\mathcal{S}} \to X^{\mathcal{S}'}$ is **a pseudo-homomorphism** between f and f' if $f = \pm f' \circ h$. The prefix "pseudo-" may be omitted if $f = f' \circ h$, and replaced by **"anti-"** if $f = -f' \circ h$. For $\mathfrak{p} = (\psi, \mathfrak{c}) \in \mathfrak{P}_{\mathcal{S}}$, write $h(\mathfrak{p}) = (h(\psi), \mathfrak{c})$.

Thus h is a pseudo-homomorphism if the following diagram commutes:

$$\begin{array}{ccc} \breve{X}^{\mathcal{S}} & \xrightarrow{\ h\ } & \breve{X}^{\mathcal{S}'} \\ {\scriptstyle f}\downarrow & & \downarrow{\scriptstyle f'} \\ \mathbb{B} & \xleftarrow[\pm]{} & \mathbb{B} \end{array}$$

Informally, h forms a pseudo-homomorphism if the value of a complete colouring on $\mathcal{S}$ can be inferred from the value of the corresponding complete colouring on $\mathcal{S}'$. Note that the function h is defined on $X^{\mathcal{S}}$ and $X^{\mathcal{S}'}$, but the requirement $f = \pm f' \circ h$ is only concerned with $\breve{X}^{\mathcal{S}}$ and $\breve{X}^{\mathcal{S}'}$, being the domains of f and f'.

[4] See Section 4.3

ψ	$f(\psi)$	$h(\psi)$	$f' \circ h(\psi)$
$(-,-,-,-)$	$-$	$(-,-,-)$	$-$
$(-,-,-,+)$	$-$	$(-,-,+)$	$-$
$(-,-,+,-)$	$-$	$(-,-,+)$	$-$
$(-,-,+,+)$	$-$	$(-,-,+)$	$-$
$(-,+,-,-)$	$-$	$(-,-,-)$	$-$
$(-,+,-,+)$	$+$	$(-,+,+)$	$+$
$(-,+,+,-)$	$+$	$(+,-,+)$	$+$
$(-,+,+,+)$	$+$	$(+,+,+)$	$+$
$(+,-,-,-)$	$-$	$(-,-,-)$	$-$
$(+,-,-,+)$	$-$	$(-,-,+)$	$-$
$(+,-,+,-)$	$+$	$(+,-,+)$	$+$
$(+,-,+,+)$	$+$	$(+,-,+)$	$+$
$(+,+,-,-)$	$-$	$(+,-,-)$	$-$
$(+,+,-,+)$	$+$	$(+,+,+)$	$+$
$(+,+,+,-)$	$+$	$(+,-,+)$	$+$
$(+,+,+,+)$	$+$	$(+,+,+)$	$+$

Table 3.4.1: All possible colourings $\psi \in \mathbb{B}^4$, showing that the function h from Example 3.4.3 is a homomorphism from f to f'. Abbreviations $-$ and $+$ are used for the Boolean values -1 and $+1$.

Definition 3.4.2. A pseudo-homomorphism h is **a pseudo-immersion** if it is injective, **a pseudo-contraction** if it is surjective, and **a pseudo-isomorphism** if it is bijective.

Example 3.4.3. Let $\mathcal{S} = \mathbb{Z}_4$, $\mathcal{S}' = \mathbb{Z}_3$, and $X = \mathbb{B}$, with

$$f : \xi \mapsto (\xi_0 \wedge \xi_2) \vee (\xi_1 \wedge \xi_2) \vee (\xi_1 \wedge \xi_3),$$
$$f' : \xi \mapsto (\xi_0 \wedge \xi_1) \vee (\xi_0 \wedge \xi_2) \vee (\xi_1 \wedge \xi_2)$$
$$h : \xi \mapsto \big((\xi_0 \wedge \xi_1) \vee (\xi_0 \wedge \xi_2) \vee (\xi_1 \wedge \xi_2), \xi_1 \wedge \xi_3, \xi_2 \vee \xi_3\big)$$

for $\psi \in \mathbb{B}^4$ and $\psi' \in \mathbb{B}^3$. Table 3.4 lists all possible colourings $\psi \in \mathbb{B}^4$, from which it can be seen that h is a homomorphism. Note that h is not a contraction since it is not surjective: the colourings $(-,-,+)$ and $(+,-,+)$ are not images of h.

A beneficial property for a homomorphism to have is the preservation of the parent-child relationship of colourings.

Definition 3.4.4. Let $h : X^{\mathcal{S}} \to X^{\mathcal{S}'}$ be a pseudo-homomorphism. Then h is **generation preserving** if

$$\psi' \leftarrow \psi \iff h(\psi') \leftarrow h(\psi)$$

for every $\psi \in X^{\mathcal{S}}$.

Lemma 3.4.5. Let $h : X^{\mathcal{S}} \to X^{\mathcal{S}'}$ be a generation preserving pseudo-homomorphism, and let $\psi \in X^{\mathcal{S}}$. Then ψ is a complete colouring if $h(\psi)$ is a complete colouring. If h is surjective then $h(\psi)$ is a complete colouring if ψ is a complete colouring.

Corollary i: If h is surjective, then $|\mathcal{U}(\psi)| = |\mathcal{U}(h(\psi))|$.

Proof. If ψ is not a complete colouring then there exists $\psi' \leftarrow \psi$, and then $h(\psi)$ is not a complete colouring since $h(\psi') \leftarrow h(\psi)$. If h is surjective then the reverse holds as well, since any child of $h(\psi)$ must be of the form $h(\psi')$ for some $\psi' \in X^{\mathcal{S}}$. The corollary then follows by induction on $|\mathcal{U}(\psi)|$, with the base case being the complete colourings. □

3.5 Complexity

Several games on propositional formulas were proved to be PSPACE-complete by Schaefer [88]. This includes the game that Schaefer calls G_ω(POS CNF), which is equivalent to isotone GAME-SAT. Thus GAME-SAT is PSPACE-complete even when restricted to isotone functions. On the other hand GAME-SAT is in PSPACE since it can be solved by a backtracking tree search algorithm whose space requirements are linear in the number of variables. Therefore GAME-SAT must be PSPACE-complete in the general case as well.

The first specific game that was proved to be PSPACE-complete was generalized Hex[5] by Even and Tarjan [29]. They observe that "any game with a sufficiently rich structure" will probably fall into this class. Informally, what makes a PSPACE-complete problem likely more difficult than an NP-complete problem is that verifying a solution to a PSPACE-complete problem is essentially as much work as solving the problem in the first place. Thus most games, as opposed to most puzzles, likely do not lie in NP.

In set colouring games as defined in this chapter it is illegal to "uncolour" or re-colour an item. If this restriction were removed, the length of the game would no longer be bounded by $|\mathcal{S}|$ or even $|\mathcal{S}| \cdot |X|$. This would make a crucial difference, as shown by Stockmeyer and Chandra who proved certain games of this kind are EXPTIME-complete [97]. Thus these games are in some sense much harder still. Some commonly played games, including chess and checkers, fall into this class when generalized to arbitrary board sizes [31, 86]. Puzzles whose solution lengths are not polynomially bounded suffer a similar fate of being in a harder class than their counterparts with bounded solution lengths; for instance, Sokoban is not in NP but is PSPACE-complete [25].

[5]See Section 4.5.

Chapter 4

Related Games

Several classes of previously studied games are special cases of set colouring games. The game itself can be generalized even further to range over some ordered set larger than $\mathbb{B}$; for instance, draws can be included by using the range $\mathbb{T}$ for the outcome. All of the theorems of Chapter 3 still hold for such **multi-valued set colouring games**. However, as will be discussed in Section 4.8, there really is no need to have more than two possible outcomes.

Since one of the main motivations behind this thesis is the game of Hex, this chapter will concentrate on a series of specializations of set colouring games that culminate in Hex.

4.1 QBF

The **Quantified Boolean Formula** (QBF) problem involves a Boolean formula preceded by quantifiers, for instance:

$$\exists_{\psi_0} \forall_{\psi_1} \exists_{\psi_2} \forall_{\psi_3} \dots \exists_{\psi_{k-1}} [f(\psi) = +1]$$

for given $f : \mathbb{B}^k \to \mathbb{B}$. Any quantified formula can be represented in QBF form by first transforming it into **prenex** form, which means that all the quantifiers are at the front. This can be done in polynomial time [27]. The quantifiers can then be made to alternate between existential and universal by inserting a quantifier with a dummy variable between each pair of non-alternating quantifiers.

QBF is the canonical PSPACE-complete problem [37]. Any QBF formula can be turned into an equivalent unquantified Boolean formula, since $\exists_\chi [f(\psi^* \chi^v)] = f(\psi^* \mathrm{T}^v) \vee f(\psi^* \mathrm{F}^v)$ and $\forall_\chi [f(\psi^* \chi^v)] = f(\psi^* \mathrm{T}^v) \wedge f(\psi^* \mathrm{F}^v)$. Thus QBFs are not more expressive than regular SAT, but in many cases the representation is more economical, as each quantifier tends to double the number of clauses. This can cause the SAT expression to become exponentially longer, causing the shift from NP-completeness to PSPACE-completeness as solutions can no longer be verified in polynomial time.

Much attention has traditionally been given to SAT solvers in the AI community, but recent years have seen an increasing interest in QBF solvers.[1] Competitions for QBF solvers now exist, much like the SAT solver competitions [87].

4.2 GAME-SAT

GAME-SAT is equivalent to set colouring games of the form $\langle \mathbb{T}^n, f \rangle$. Games on Boolean functions or propositional formulas were studied by Schaefer [88] and Stockmeyer and Chandra [97], who obtained complexity results for several classes of games. In Artificial Intelligence, GAME-SAT was introduced by Zhao and Müller [107] in the context of expressing dependencies between subgoals in planning for the game of Go.

GAME-SAT has obvious similarities with QBF, as the latter can be seen as a game where MAX assigns the elements tied to universal quantifiers, and MIN assigns the elements tied to existential quantifiers. The difference with GAME-SAT is that the QBF players are not free to choose which element to colour next. Since both games are PSPACE-complete there must exist polynomial-time reductions between them, though no specific reduction has yet been demonstrated. To encode $n \times n$ Hex in QBF form, one could use the expression $\forall_{\mathfrak{m}_1} \exists_{\mathfrak{m}_2} \forall_{\mathfrak{m}_3} \dots f(\mathfrak{m}_1, \mathfrak{m}_2, \dots)$ where the $\mathfrak{m}_i$ represent the moves, but as each move involves a choice between $\mathcal{O}(n^2)$ cells it needs $\mathcal{O}(\log n)$ binary variables to encode. Furthermore it complicates the outcome function, which needs to "reconstruct" the board position from the binary encodings of the moves. The increase in representation size is sublinear, but the GAME-SAT representation is considerably more economical in practice.

The existing QBF competitions all concentrate on *solving* QBF problems, which in game terms amounts to "ultra-weakly solving" a game, namely determining the player who has a winning strategy from the opening position. A harder problem in game theory is "weakly solving" a game, which involves determining an explicit winning strategy from the opening position, or even "strongly solving" a game, which means being able to find correct moves from *any* position. When none of these are feasible with realistic resources, then the problem becomes heuristic: How to play *well* when one cannot play perfectly. Standard QBF competitions do not address these game-perspective issues.

4.3 Division Games

The class of **division games** was introduced by Yamasaki [102]. Division games are essentially equivalent to GAME-SAT with the extra requirement that MAX may only assign the value +1 to elements, and MIN may only assign the value -1. In such a case it makes sense to speak of a player "owning" or "occupying" a element. It then also makes sense to speak of increasing elements as "regular", and of decreasing elements as "misère" elements, indicating respectively that one prefers to own or disown them.

[1] See Section 11.4.

If there are no misère elements, a division game is just an isotone GAME-SAT instance, and vice versa. If however there are misère elements, the game is radically different. Yamasaki further expanded his theory to allow games in which the players do not simply alternate moves, but get to play moves according to a predefined "schedule". See Section 4.8 for a discussion on these topics.

4.4 Coalition Games

A special case of division games and GAME-SAT is the coalition game, where there is a specified list of subsets of the variables. Player MAX tries to occupy all the elements of at least one of the subsets. Player MIN, therefore, tries to occupy at least one element in every subset. These games were described in Section 2.4. Each coalition game with scoring function $\text{COAL}(\mathcal{F}, \{\text{TRUE}\})$ has a dual representation with scoring function $\text{COAL}(\mathcal{F}', \{\text{FALSE}\})$, and by Theorem 2.4.4 any coalition function is an isotone Boolean function and vice versa. Thus all theorems for isotone set colouring games apply.

4.5 Shannon Switching Games

The Shannon switching game is played on any finite graph $\mathcal{G}$ with two distinguished vertices. The distinguished vertices are called **terminals**. The two players take turns colouring the edges of the graph with two colours, say blue and red. The goal for MAX is to connect the terminals with a blue path. The goal for MIN is to avoid this. In Shannon switching games, the players MAX and MIN are often referred to as "Short" and "Cut".

This game is a coalition game, played on $\mathcal{E}(\mathcal{G})$ where a subset of $\mathcal{E}(\mathcal{G})$ is a coalition if and only if it contains an inter-terminal path. It is therefore an isotone GAME-SAT instance. A polynomial-time algorithm for recognizing winning and losing positions was discovered by Lehman [62], generalizing an earlier solution found by Oliver Gross for the game **Bridg-It** which was introduced by Gale [33, 35].

The Shannon switching game can be modified by requiring the players to colour the graph's vertices rather than its edges. The vertex colouring game is in some respects more interesting. Edge colouring is indeed a special case of vertex colouring, as playing the edge colouring game on a graph $\mathcal{G}$ is equivalent to playing the vertex colouring game on $\mathcal{G}$'s line graph. Moreover, the vertex colouring game is likely fundamentally more complex, as it has been shown to be PSPACE-complete [29]. Unless otherwise specified, in the remainder of the text the **Shannon game** shall refer to the vertex colouring game.

By convention, in the Shannon game the two terminal vertices shall be coloured blue before the game starts, to remove any confusion. The **internal** vertices are all the non-terminal ones. The winning condition can equivalently be stated as saying that Short wins if the terminals are adjacent after contracting all blue internal vertices, and Cut wins if the terminals are disconnected after removing all red vertices.

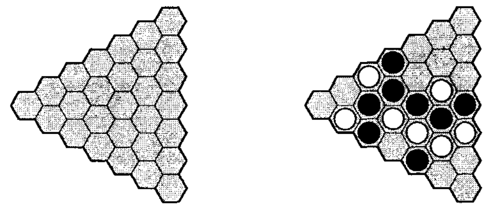


Figure 4.1: Empty Y-board (left), and a win for Black (left).

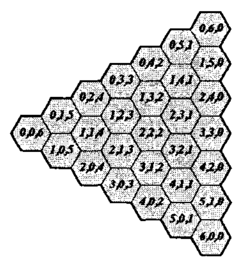


Figure 4.2: Coordinate numbering for Y boards.

This game is an isotone GAME-SAT instance. As will be discussed in Section 5.2, any sensible players will only want to use their own colours, so the rules need not specify this restriction. In the coalition formulation of the game the coalitions are the vertex sets that connect the terminals. In the dual representation of the game the coalitions are the cut sets that disconnect the terminals.

4.6 Game of Y

The game of Y was introduced by Milnor and Shannon [71, 34] and independently Schensted [91] in the early 1950s. A Y board consists of a triangular configuration of hexagons. Players take turns colouring the hexagons. White's goal is to construct a white chain that touches all three sides of the board. Black's goal is to achieve the same with a black chain. Figure 4.1 shows an empty Y board, and a Y board containing a winning chain for Black.

Let $\mathbb{Y}_n$ represent the n-sided Y board, so that $|\mathbb{Y}_n| = \frac{1}{2}n(n+1)$, with elements numbered as in Figure 4.2. This system uses three redundant indices (x, y, z) with $x + y + z = n - 1$; each index encodes the distance to one of the three edges of the board.[2] The distance between the cells (x, y, z) and (x', y', z') is $\frac{1}{2}(|x - x'| + |y - y'| + |z - z'|)$. The corresponding scoring function for $\mathbb{Y}_n$ is $f_{\mathbb{Y}n} : \mathbb{B}^{\mathbb{Y}_n} \to \mathbb{B}$ which is defined on all complete colourings of $\mathbb{Y}_n$.

[2]These indices start with 0, contrary to the coordinates on a Hex board. The reason for this is related to practical implementation matters: the rows and columns 0 and $n+1$ on $\mathbb{X}_n$ are reserved for the border pieces (see Section 2.6). On $\mathbb{Y}_n$ this is not necessary as the borders are not associated with the players.

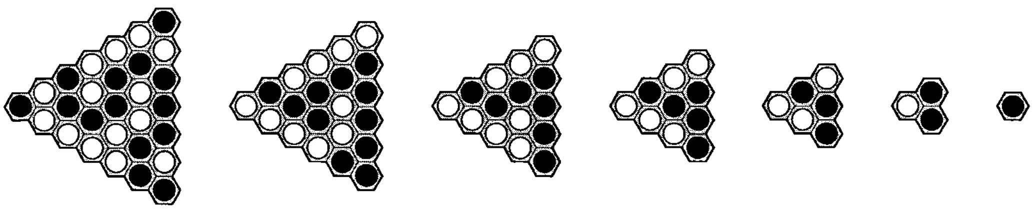


Figure 4.3: A chain of Y reductions.

The game of Y shares the property of Hex that any completely coloured board will contain a winning group for exactly one of the players, but not both. A remarkable proof of this assertion is based on an observation by Schensted, and given previously in [48]. Consider the function $h : \mathbb{B}^{\mathbb{Y}_n} \longrightarrow \mathbb{B}^{\mathbb{Y}_{n-1}}$ defined as follows:

$$h(\psi) : (x, y, z) \mapsto (\psi_{x+1,y,z} \wedge \psi_{x,y+1,z}) \vee (\psi_{x+1,y,z} \wedge \psi_{x,y,z+1}) \vee (\psi_{x,y+1,z} \wedge \psi_{x,y,z+1})$$

for $\psi \in \mathbb{B}^{\mathbb{Y}_n}$. Informally this means that the value of $h(\psi)_{x,y,z}$ is the "majority vote" of the three "surrounding" elements in ψ. This function is called the **Y reduction**. Figure 4.3 shows a chain of such Y reductions, ending with a position on $\mathbb{Y}_1$.

Y reduction is a contraction from $f_{\mathbb{Y}n}$ to $f_{\mathbb{Y}n-1}$; in other words,

$$f_{\mathbb{Y}n}(\psi) = +1 \Longleftrightarrow f_{\mathbb{Y}n-1}(h(\psi)) = +1.$$

To prove '$\Longrightarrow$', let $\big((x_0, y_0, z_0), (x_1, y_1, z_1), \ldots, (x_{k-1}, y_{k-1}, z_{k-1})\big)$ be a sequence of coordinates that encodes a winning chain for MAX, so that:

1. $\forall_{i \in \mathbb{Z}_k} \big[\psi(x_i, y_i, z_i) = +1\big]$;
2. $\exists_{i \in \mathbb{Z}_k} [x_i = 0]$ and $\exists_{i \in \mathbb{Z}_k} [y_i = 0]$ and $\exists_{i \in \mathbb{Z}_k} [z_i = 0]$;
3. $\forall_{i \in \mathbb{Z}_{k-1}} \big[\frac{1}{2}(|x_i - x_{i+1}| + |y_i - y_{i+1}| + |z_i - z_{i+1}|) = 1\big]$.

In other words, the chain belongs to MAX, touches all three sides, and is connected. Now put $\psi' = h(\psi)$ and $(x'_i, y'_i, z'_i) = (\min[x_i, x_{i+1}], \min[y_i, y_{i+1}], \min[z_i, z_{i+1}])$ for all $i \in \mathbb{Z}_{k-2}$. Then it can be verified that:

1. For all $i \in \mathbb{Z}_{k-1}$ we have $\psi'(x_i, y_i, z_i) = +1$ since both (x_i, y_i, z_i) and $(x_{i+1}, y_{i+1}, z_{i+1})$ are among the three "surrounding" elements in ψ;
2. take the first index i for which $x_i = 0$, then $x'_i = \min[x_i, x_{i+1}] = 0$, and similarly $\exists_{i \in \mathbb{Z}_{k-1}} [y'_i = 0]$ and $\exists_{i \in \mathbb{Z}_{k-1}} [z'_i = 0]$;
3. since $x'_i, x'_{i+1} \in \{x_{i+1} - 1, x_{i+1}\}$ we have $|x'_i - x'_{i+1}| \leqslant 1$ so that $\frac{1}{2}(|x'_i - x'_{i+1}| + |y'_i - y'_{i+1}| + |z'_i - z'_{i+1}|) \leqslant \frac{3}{2}$ which means the distance between (x'_i, y'_i, z'_i) and $(x'_{i+1}, y'_{i+1}, z'_{i+1})$ must be at most equal to 1.

Therefore $\big((x'_0, y'_0, z'_0), (x'_1, y'_1, z'_1), \ldots, (x'_{k-2}, y'_{k-2}, z'_{k-2})\big)$ is a winning chain for MAX as well.

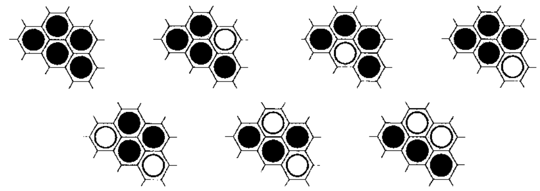


Figure 4.4: The seven symmetrically distinct patterns that yield two connected black cells after Y reduction.

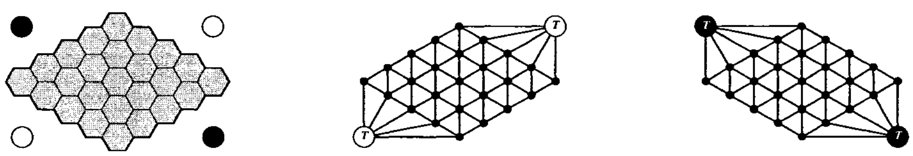


Figure 4.5: Dual representations of 5 × 5 Hex as a Shannon game, with White (middle) or Black (right) playing the role of Short.

This suffices to prove that *at most* one player can have a winning chain on $\mathbb{Y}_n$, since player $\mathfrak{c}$ has a winning chain on $\mathbb{Y}_1$ if $\mathfrak{c}$ has a winning chain on $\mathbb{Y}_n$. To prove that h is a proper contraction, and that therefore *exactly* one player has a winning chain on $\mathbb{Y}_n$, it suffices to show additionally that $f_{\mathbb{Y}n}(\psi) = +1 \Longleftarrow f_{\mathbb{Y}n-1}(h(\psi)) = +1$. Consider two consecutive elements from a winning chain for $\mathfrak{c}$ on $\mathbb{Y}_{n-1}$. These correspond to two touching triangles on $\mathbb{Y}_n$, each of which contains at least two cells belonging to $\mathfrak{c}$. An enumeration of cases shows that these two triangles must form one of the patterns shown in Figure 4.4, containing one single connected group belonging to $\mathfrak{c}$. Thus the entire $\mathfrak{c}$-chain on $\mathbb{Y}_{n-1}$ must correspond to a connected $\mathfrak{c}$-chain on $\mathbb{Y}_n$. Each cell in the chain on $\mathbb{Y}_{n-1}$ that touches one of the sides corresponds to a triangle on $\mathbb{Y}_n$ that touches the same side, and since at least two of the cells in that triangle belong to $\mathfrak{c}$, at least one of those $\mathfrak{c}$-cells touches the same side. Therefore the $\mathfrak{c}$-chain on $\mathbb{Y}_n$ also touches all three sides, and is a winning chain for $\mathfrak{c}$.

4.7 Hex

Hex can be represented both as a Shannon game as well as a Y game. Figure 4.5 displays two alternative representations of 5 × 5 Hex as a Shannon game. Both graphs are created from the adjacency graph of the Hex board cells, with the addition of two terminal vertices labelled 'T'. Any path that connects the two terminals must contain a neighbour of each terminal, and as can be seen in Figure 4.5 the terminal neighbours are exactly the groups of cells that are to be connected by White, in the middle diagram, or Black, in the right diagram. This method can be used in general to generate a Shannon game graph for any game where one player tries to connect two specified groups of cells or vertices.

Schensted pointed out that Hex is also a special case of the game of Y because a Y position can be

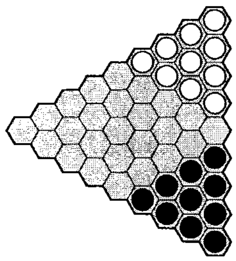


Figure 4.6: Representation of 5 × 5 Hex as a size-9 Y game.

set up in which each player wins the Y game by winning the Hex game in the empty region, and vice versa [91]. Figure 4.6 shows such a position.[3] If White were to connect the two "Hex borders" in the interior region, the connecting chain would touch the lower left edge of the Y board, as well as the upper right group of White stones, through which it would touch the other two edges of the Y board. The dual argument applies to Black, and so the winner of the Hex game equals the winner of the Y game.

4.8 Limitations of Set Colouring Games

Whereas set colouring games form a general class modelling a variety of games, there are classes of games that are quite similar yet violate certain set colouring game rules. Some of these games can still be modelled as set colouring games, or possibly multi-valued set colouring games, by straightforward transformations. In other cases the nature of the game changes too drastically for this to be possible.

Undoing or Skipping Moves

Allowing players to skip a move alters the strategy of the game. However, a skip-legal game may still be modelled as a set colouring game by adding a suitably large number of dead elements to the game. Indeed, according to Theorem 5.4.2, adding *one* dead element suffices.

As mentioned in Section 3.5, allowing players to "uncolour" an element, or colour an already coloured element, alters the nature of the game in such a fundamental manner that the game is generally not even in the same complexity class anymore. Therefore such games cannot be modelled using the definitions in this thesis.

[3]Schensted uses a slightly different diagram, containing only the pieces immediately adjacent to the "Hex region". This is equivalent to the Y position in Figure 4.6 since the extra pieces near the corners of the Y board are dead.

Restricted Moves

Players may be restricted in their choice of element to colour. The subset of elements that a player is allowed to assign may be fixed throughout the game, or may depend on the position in some way. Examples of the latter case are connect-four, where no move may have an empty cell underneath it, and renju, where the first player is forbidden to form certain patterns. Another example is the QBF problem itself, where the order of assignment of the elements is specified in advance; this is implicitly a position-dependent element choice since the next element to be assigned can be deduced from the position.

Restricted Colours

Apart from the choice of element, there may be restrictions on the colour to assign to it. If these restrictions are independent of the position and the player, then the game can still be modelled as a set colouring game. Say that each element v has its own colouring domain X_v, then a regular set colouring game can be constructed with $X = \bigcup_{v \in \mathcal{S}} X_v$. For each v pick a representing colour $\chi_v \in X_v$; the outcome of a complete colouring ψ is then obtained by replacing ψ_v by χ_v whenever $\psi_v \notin X_v$. The resulting game meets the specifications of set colouring games, and exhibits identical strategic behaviour.

The most common case of colour restriction is where players must use their "own" colour; in GAME-SAT terms this means that MAX may only assign the value TRUE and MIN may only assign the value FALSE. This makes no strategic difference if the game is played on an isotone function, since it would be irrational to do otherwise even if it were allowed. However, if the function is not isotone then this restriction fundamentally alters the nature of the game.

For instance, in the game of "misère Hex" the usual goals for the players are reversed. This is not the same as negating the outcome function, as that would merely achieve a switching of roles between the two players. It is known that misère Hex is a loss for the *last* player to move; in other words, it is a win for the first player to move if and only if the board size is even [59]. This means that the correct strategy for playing misère Hex is not merely to play moves that would lose in regular Hex, since opening positions on odd-sized boards larger than 1×1 contain losing moves in regular Hex but no winning moves in misère Hex. Indeed the strategy for misère Hex is radically different.

What differentiates these games from set colouring games is that they are **partizan games**, as opposed to **impartial games**. In an impartial game both players have the same moves to choose from, which in terms of set colouring games translates into position $(\psi, \mathfrak{c})$ having the same children as $(\psi, \bar{\mathfrak{c}})$. In a partizan game this is not the case. Schaefer remarks that it tends to be much easier to prove PSPACE-completeness for an impartial game than for a similar partizan game [88].

Multiple Outcomes

Many commonly played games do not feature competing Boolean goals, but a scoring function. Such a game is **a multi-valued game**. The set of possible outcomes must be totally ordered in order to make sense of optimal play, for if there were two incomparable outcomes it would be undefined what a player should do when faced with a choice between those two outcomes. Introducing an additional rule to guide such decisions amounts to imposing an ordering after all.

A multi-valued game is **zero-sum** if the outcome for one player is always exactly the negative of the outcome for the other player. More generally one can have a **constant sum** game, or simply any game in which the players have the exact opposite ordering of preference of outcomes. In all of these cases, maximizing one's outcome is equivalent to minimizing the opponent's outcome. Any multi-valued zero-sum set colouring game can be studied as a series of two-valued set colouring games, since for any possible outcome t one may ask the Boolean question whether or not MAX can achieve at least t.

Ambiguous Final Positions

Instead of using a multi-valued scoring function, many other commonly played games are defined in terms of *two* goals, one for each player. This is the case in the traditional formulation of Hex. It may then turn out, as it does in Hex, that the two goals are exactly each other's negation. In other games, however, it may be possible that neither goal ends up being fulfilled, or that both goals are. The natural convention when neither goal is fulfilled is to declare the game a draw. If this possibility exists then the game is a multi-valued set colouring game where the outcome function ranges over $\mathbb{T}$, and can therefore be studied using the methods for Boolean set colouring games.

If it is possible that both goals are fulfilled, the winner traditionally is the *first* player to do so. Such a **race game** cannot be modelled as a set colouring game or even a multi-valued set colouring game, since the winner of the game cannot be determined from the game's final position when play continues until all elements are coloured. Well-known examples are n-in-a-row games such as tic-tac-toe, go-moku, and qubic.

Non-Alternating Turns

In all games described thus far, the two players alternate turns. Commonly played games rarely violate this principle. Some results are informally known for **equalized Hex**, where the first player plays one piece as the opening move and thereafter both players play *two* pieces on each subsequent turn. This game was shown to be a first player win on the 5×5 board by Bush, Heuer, and Huddleston [21]. In the class of **division games**, introduced and described by Yamasaki [102], there is a predetermined but not necessarily alternating order of turns. This order is a vector $\underline{w} \in \mathfrak{C}^n$ where $\underline{w}_i$ indicates the player to move at turn i. Yamasaki's key theorem states that the last player to move in order $\underline{w}$ is at least as well off in the game with order $\underline{w}'$ when $\underline{w}'_i = \underline{w}_{i+1 \pmod n}$.

This implies the first-player-win property of Hex. Yamasaki's definition of division games also differs from set colouring games in that players are forced to use only their own colour.

Alternatively, the order may be determined stochastically, as in **coinflip Hex** where each move is preceded by a coin toss to appoint the next player to move. Peres *et al* proved that the probability of winning such a coinflip coalition game with optimal play is equal to the probability of winning the corresponding alternating turn game when playing randomly [78]. Optimal moves for such games can be approximated by random sampling. This result is closely connected to the Monte Carlo evaluation methods to be described in Section 12.5. In coinflip coalition games the optimal element to colour is always the same for both players.

The definition of the minimax function can be modified to fit both non-alternating turn games and stochastic turn games in a straightforward manner. The resulting strategy can be radically different from the strategy in the alternating turn variant, as is the case for equalized Hex and coinflip Hex.

Chapter 5

Minimax Values

Section 3.3 defined strategies and the criteria for a winning strategy. A strategy is said to be optimal if it guarantees the best possible result against any strategy chosen by the opponent. This best possible result is the **game theoretical value** of a game, and it can be computed recursively using the minimax function.

5.1 The Minimax Function

In Definition 3.2.3, Boolean functions on $\breve{X}^{\mathcal{S}}$ were extended to Boolean functions on final positions in $\breve{X}^{\mathcal{S}} \times \mathfrak{C}$. These functions can be further extended to all of $X^{\mathcal{S}} \times \mathfrak{C}$, representing the best outcome that each player can guarantee against any possible strategy by the opponent. The following recursive definition corresponds to the standard definition in game theory.

Definition 5.1.1. Let $\langle X^{\mathcal{S}}, f\rangle$ be a set colouring game and $\mathfrak{p} = (\psi, \mathfrak{c}) \in \mathfrak{P}_{\mathcal{S}}$. The **minimax value** $\textsc{mnx}(f; \mathfrak{p})$ is defined as:

$$\textsc{mnx}(f;\mathfrak{p}) \stackrel{\text{def}}{=\!=} \textsc{mnx}(f;\psi,\mathfrak{c}) \stackrel{\text{def}}{=\!=} \begin{cases} f(\psi) & \text{if } \mathfrak{p} \text{ is a final position,} \\ \min_{\xi \leftarrow \mathfrak{p}}\left[\textsc{mnx}(f;\xi)\right] & \text{if } \mathfrak{c} = \textsc{min}, \\ \max_{\xi \leftarrow \mathfrak{p}}\left[\textsc{mnx}(f;\xi)\right] & \text{if } \mathfrak{c} = \textsc{max}. \end{cases}$$

For $\Gamma = \langle X^{\mathcal{S}}, f\rangle$ the notation $\textsc{mnx}(\Gamma;\mathfrak{p})$ refers to $\textsc{mnx}(f;\mathfrak{p})$, and $\textsc{mnx}(\Gamma;\mathfrak{c})$ is shorthand for the value $\textsc{mnx}(\Gamma;\phi^{\mathcal{S}},\mathfrak{c})$.

The minimax value is uniquely determined for each position, and displays the properties listed in the following theorem:

Theorem 5.1.2. Let $\langle X^{\mathcal{S}}, f\rangle$ and $\langle X^{\mathcal{S}}, f'\rangle$ be two set colouring games, and let $\mathfrak{p} = (\psi, \mathfrak{c}) \in \mathfrak{P}_{\mathcal{S}}$. Then:

i. If $\forall_{\psi^* \trianglerighteq \psi}\left[f(\psi^*) \geqslant f'(\psi^*)\right]$ then $\textsc{mnx}(f;\mathfrak{p}) \geqslant \textsc{mnx}(f';\mathfrak{p})$;

ii. If $\forall_{\psi^* \trianglerighteq \psi}\left[f(\psi) \geqslant t\right]$ for some $t \in \mathbb{B}$, then $\textsc{mnx}(f;\mathfrak{p}) \geqslant t$.

These assertions of course also hold with $\leqslant$ or $=$ substituted for $\geqslant$.

The **minimax theorem** [73] states that

$$\textsc{mnx}(f;\mathfrak{p}) = \min_{\mathfrak{s}_{\text{MIN}} \in \mathfrak{S}_{\mathcal{S}}}\left[\max_{\mathfrak{s}_{\text{MAX}} \in \mathfrak{S}_{\mathcal{S}}}\left[f(\mathfrak{p}, \mathfrak{s}_{\text{MIN}} * \mathfrak{s}_{\text{MAX}})\right]\right] = \max_{\mathfrak{s}_{\text{MAX}} \in \mathfrak{S}_{\mathcal{S}}}\left[\min_{\mathfrak{s}_{\text{MIN}} \in \mathfrak{S}_{\mathcal{S}}}\left[f(\mathfrak{p}, \mathfrak{s}_{\text{MIN}} * \mathfrak{s}_{\text{MAX}})\right]\right]$$

which means that the minimax value of a position is the best outcome that each player can guarantee against any possible opposing strategy.

Definition 5.1.3. Let $\langle X^{\mathcal{S}}, f\rangle$ be a set colouring game and $\mathfrak{p} = (\psi, \mathfrak{c}) \in \mathfrak{P}_{\mathcal{S}}$. The **negamax value** of $\mathfrak{p}$ is defined as $\textsc{ngx}(f;\mathfrak{p}) \stackrel{\text{def}}{=\!=} \lambda(\mathfrak{c}) \cdot \textsc{mnx}(f;\mathfrak{p})$. The notations $\textsc{ngx}(\Gamma;\mathfrak{p})$ and $\textsc{ngx}(\Gamma;\mathfrak{c})$ are used analogously to $\textsc{mnx}(\Gamma;\mathfrak{p})$ and $\textsc{mnx}(\Gamma;\mathfrak{c})$.

Observation i: $\textsc{ngx}(f;\mathfrak{p}) = \max_{\xi \leftarrow \mathfrak{p}}\left[-\textsc{ngx}(f;\xi)\right] = -\min_{\xi \leftarrow \mathfrak{p}}\left[\textsc{ngx}(f;\xi)\right]$.

Observation ii: If $\mathfrak{M}(\mathfrak{p}) \neq \varnothing$ then $\textsc{ngx}(f;\mathfrak{p}) = +1 \iff \exists_{\mathfrak{m} \in \mathfrak{M}(\mathfrak{p})}\left[\textsc{ngx}(f;\mathfrak{p} \oplus \mathfrak{m}) = -1\right]$.

Observation iii: If $\mathfrak{M}(\mathfrak{p}) \neq \varnothing$ then $\textsc{ngx}(f;\mathfrak{p}) = -1 \iff \forall_{\mathfrak{m} \in \mathfrak{M}(\mathfrak{p})}\left[\textsc{ngx}(f;\mathfrak{p} \oplus \mathfrak{m}) = +1\right]$.

Observation iv: $\textsc{ngx}(f;\mathfrak{p}) \geqslant -\textsc{ngx}(f;\mathfrak{p} \oplus \mathfrak{m})$ for any $\mathfrak{m} \in \mathfrak{M}(\mathfrak{p})$.

Informally, the minimax function answers the question "can MAX force a win?" and the negamax function answers the question "can the next player to move force a win?" The two functions lead to the following concepts:

Definition 5.1.4. Let $\Gamma = \langle X^{\mathcal{S}}, f\rangle$, $\mathfrak{p} \in \mathfrak{P}_{\mathcal{S}}$, and $\mathfrak{m} \in \mathfrak{M}(\mathfrak{p})$.

- **Winning**: $\mathfrak{m}$ is Γ-winning in $\mathfrak{p}$ if $\textsc{ngx}(f;\mathfrak{p} \oplus \mathfrak{m}) = -1$. The set of Γ-winning moves in $\mathfrak{p}$ is denoted $\mathfrak{M}^{+}_{\Gamma}(\mathfrak{p})$.
- **Losing**: $\mathfrak{m}$ is Γ-losing in $\mathfrak{p}$ if $\textsc{ngx}(f;\mathfrak{p} \oplus \mathfrak{m}) = +1$. The set of Γ-losing moves in $\mathfrak{p}$ is denoted $\mathfrak{M}^{-}_{\Gamma}(\mathfrak{p})$.
- **Optimal**: $\mathfrak{m}$ is Γ-optimal in $\mathfrak{p}$ if $\textsc{mnx}(f;\mathfrak{p} \oplus \mathfrak{m}) = \textsc{mnx}(f;\mathfrak{p})$. The set of Γ-optimal moves in $\mathfrak{p}$ is denoted $\mathfrak{M}^{o}_{\Gamma}(\mathfrak{p})$.
- A strategy $\mathfrak{s}$ is Γ-optimal for player $\mathfrak{c}$ if $\mathfrak{s}(\psi) \in \mathfrak{M}^{o}_{\Gamma}(\psi, \mathfrak{c})$ for every $\psi \in X^{\mathcal{S}}$.

With these definitions, the following properties are easily verified for any non-final position $\mathfrak{p} \in \mathfrak{P}_{\mathcal{S}}$ and any move $\mathfrak{m} \in \mathfrak{M}(\mathfrak{p})$:

- $\mathfrak{m} \in \mathfrak{M}^{o}_{\Gamma}(\mathfrak{p}) \iff \textsc{ngx}(f;\mathfrak{p}) = -\textsc{ngx}(f;\mathfrak{p} \oplus \mathfrak{m})$;
- $\mathfrak{M}^{+}_{\Gamma}(\mathfrak{p})$ and $\mathfrak{M}^{-}_{\Gamma}(\mathfrak{p})$ form a partition of $\mathfrak{M}(\mathfrak{p})$;

positive	negative	regular	misère
TRUE	FALSE	ξ_0	
$\xi_0 \vee \xi_1$	$\xi_0 \wedge \xi_1$	$\xi_0 \equiv \xi_1$, n *odd*	$\xi_0 \equiv \xi_1$, n *even*
		$\xi_0 \vee (\xi_1 \wedge \xi_2)$	
		$\xi_0 \wedge (\xi_1 \vee \xi_2)$	

Table 5.1.1: Examples of set colouring games $\langle \mathbb{T}^n, f\rangle$ classified according to Definition 5.1.5; entries represent $f : \xi \mapsto \ldots$ for suitably large n.

- $\mathrm{NGX}(f; \mathfrak{p}) = +1 \iff \mathfrak{M}^{\circ}_{\Gamma}(\mathfrak{p}) = \mathfrak{M}^{+}_{\Gamma}(\mathfrak{p}) \neq \varnothing$;
- $\mathrm{NGX}(f; \mathfrak{p}) = -1 \iff \mathfrak{M}^{\circ}_{\Gamma}(\mathfrak{p}) = \mathfrak{M}^{-}_{\Gamma}(\mathfrak{p}) = \mathfrak{M}(\mathfrak{p}) \iff \mathfrak{M}^{+}_{\Gamma}(\mathfrak{p}) = \varnothing$;
- $\mathfrak{M}^{\circ}_{\Gamma}(\mathfrak{p}) \neq \varnothing$;

By definition, when both players only play optimal moves starting in $\mathfrak{p}$, then the outcome of the game is $\mathrm{MNX}(f; \mathfrak{p})$. Optimal play implicitly assumes that the opponent is also playing optimally. An optimal strategy cannot be exploited, but it cannot itself exploit fallible opponents either [15]. If the opponent is fallible then optimal play guarantees an outcome at least as desirable as the optimal outcome, though not necessarily the *best* outcome that can be achieved against the particular opponent in question. To optimize the expected result against a non-optimal opponent a model is needed to approximate the opponent's behaviour [53].

Definition 5.1.5. Let $\Gamma = \langle X^{\mathcal{S}}, f\rangle$. A position $\mathfrak{p} \in \mathfrak{P}_{\mathcal{S}}$ is **a win** in Γ if $\mathrm{NGX}(\Gamma; \mathfrak{p}) = +1$, and **a loss** otherwise. For a colouring $\psi \in X^{\mathcal{S}}$:

ψ is	if	for every $\mathfrak{c} \in \mathfrak{C}$
regular		$\mathrm{NGX}(\Gamma; \psi, \mathfrak{c}) = +1$
misère		$\mathrm{NGX}(\Gamma; \psi, \mathfrak{c}) = -1$
positive		$\mathrm{MNX}(\Gamma; \psi, \mathfrak{c}) = +1$
negative		$\mathrm{MNX}(\Gamma; \psi, \mathfrak{c}) = -1$

Each colouring belongs to exactly one of these four categories. The same terminology is applied to Γ itself by taking $\psi = \phi^{\mathcal{S}}$.

Some examples are listed in Table 5.1.1.

5.2 Rational Moves

In some cases one of the players, or both, may have a clear preference which colour to assign to a certain element, regardless of the rest of the position. This will lead to the notion of "optimal colourings" and "rational moves", which will eventually be extended to sets of elements and to "superrational play" in Chapter 8.

Definition 5.2.1. Let $\Gamma = \langle X^{\mathcal{S}}, f\rangle$, $v \in \mathcal{S}$, and $\mathfrak{m}^-, \mathfrak{m}^+ \in \mathfrak{M}(v)$. If $f(\psi^*\mathfrak{m}^+) \geqslant f(\psi^*\mathfrak{m}^-)$ for all $\psi^* \in \check{X}^{\mathcal{S}}$, then $\mathfrak{m}^+$ is **preferable** to $\mathfrak{m}^-$ for MAX, and $\mathfrak{m}^-$ is preferable to $\mathfrak{m}^+$ for MIN.

> *Observation i:* Let $\check{X}$ be ordered and let $\chi^-, \chi^+ \in \check{X}$ with $\chi^+ \geqslant \chi^-$. Then $(\chi^+)^v$ is preferable to $(\chi^-)^v$ for MAX if f is increasing in v, and for MIN if f is decreasing in v.

Preferable colourings are so called because they do lead to comparisons where one position is preferable to another for one of the players.

Theorem 5.2.2. Let $\Gamma = \langle X^{\mathcal{S}}, f\rangle$, $\mathfrak{p} \in \mathfrak{P}_{\mathcal{S}}$, and $v \in \mathcal{S}$. Let $\mathfrak{m}^-, \mathfrak{m}^+ \in \mathfrak{M}(v)$ with $\mathfrak{m}^+$ preferable to $\mathfrak{m}^-$ for MAX. Then $\text{MNX}(f; \mathfrak{p}\mathfrak{m}^+) \geqslant \text{MNX}(f; \mathfrak{p}\mathfrak{m}^-)$.

> *Corollary i:* If $v \in \mathcal{U}(\mathfrak{p})$ then $\text{MNX}(f; \mathfrak{p} \oplus \mathfrak{m}^+) \geqslant \text{MNX}(f; \mathfrak{p} \oplus \mathfrak{m}^-)$.
>
> *Corollary ii:* If v is dead in f then $\text{MNX}(f; \mathfrak{p}\mathfrak{m}')$ is constant for every $\mathfrak{m}' \in \mathfrak{M}(v)$.

From this we get the definition of a rational move.

Definition 5.2.3. Let $\Gamma = \langle X^{\mathcal{S}}, f\rangle$, $\mathfrak{p} = (\psi, \mathfrak{c}) \in \mathfrak{P}_{\mathcal{S}}$, $\mathfrak{c} \in \mathfrak{C}$, and $\mathfrak{m} = \chi^v \in \mathfrak{M}(\mathfrak{p})$. If $\mathfrak{m}$ is preferable for $\mathfrak{c}$ to every $\mathfrak{m}' \in \mathfrak{M}(v)$, then $\mathfrak{m}$ is **rational** for $\mathfrak{c}$ and **irrational** for $\bar{\mathfrak{c}}$.

> *Observation i:* Element v is dead in f if and only if any move in $\mathfrak{M}(v)$ is rational for both players.
>
> *Observation ii:* Let $\check{X}$ be ordered, then T^v is rational for MAX and F^v is rational for MIN if f is increasing in v, and vice versa if f is decreasing in v.

With this terminology, Theorem 5.2.2 states that any rational move χ^v is at least as good as any other move in v, and any irrational move in v is at most as good as any other move in v. It is possible that v has a rational move for one player but not for the other player, but this can only happen if v is nonmonotone and $\check{X} > 2$.

A stronger statement than Theorem 5.2.2 can be made: Any rational move is not only preferable to an irrational move in the same element, but in fact preferable to *any* irrational move.

Theorem 5.2.4. Let $\Gamma = \langle X^{\mathcal{S}}, f\rangle$, $\mathfrak{p} = (\psi, \mathfrak{c}) \in \mathfrak{P}_{\mathcal{S}}$, and $\mathfrak{m}^+, \mathfrak{m}^- \in \mathfrak{M}(\mathcal{S})$. If $\mathfrak{m}^+$ is rational for MAX and $\mathfrak{m}^-$ is rational for MIN then $\text{MNX}(f; \mathfrak{p}\mathfrak{m}^+) \geqslant \text{MNX}(f; \mathfrak{p}\mathfrak{m}^-)$.

> *Corollary i:* $\text{MNX}(f; \mathfrak{p} \oplus \mathfrak{m}^+) \geqslant \text{MNX}(f; \mathfrak{p} \oplus \mathfrak{m}^-)$.

One piece of advice to be obtained from this is that *any* rational move is better than a dead move, and a dead move is better than any irrational move:

Theorem 5.2.5. Let $\Gamma = \langle X^{\mathcal{S}}, f\rangle$ and $\mathfrak{p} = (\psi, \mathfrak{c}) \in \mathfrak{P}_{\mathcal{S}}$. Let $\mathfrak{m}, \chi^w \in \mathfrak{M}(\mathfrak{p})$ where w is dead in f. If $\mathfrak{m}$ is rational for $\mathfrak{c}$ then $\text{NGX}(f; \mathfrak{p} \oplus \mathfrak{m}) \leqslant \text{NGX}(f; \mathfrak{p} \oplus \chi^w)$. If $\mathfrak{m}$ is irrational for $\mathfrak{c}$ then $\text{NGX}(f; \mathfrak{p} \oplus \mathfrak{m}) \geqslant \text{NGX}(f; \mathfrak{p} \oplus \chi^w)$

From this, conclude that the existence of one Γ-optimal dead move implies that all rational moves are Γ-optimal, and that there can be no Γ-winning dead move if there is no Γ-winning rational move. This means that if the live rational moves have all been found to be losses, then the dead moves do not need to be checked for they are guaranteed to lose as well.

One may now conjecture a theorem to the effect that rational moves are always at least as good as moves in non-monotone elements; in other words, "defer commitments as long as possible". This is however *not* true in general. For instance, in the game $\langle \mathbb{T}^5, \xi \mapsto (\xi_0 \equiv (\xi_1 \vee \xi_2)) \wedge (\xi_3 \vee \xi_4)\rangle$ with MAX to move first, the only rational moves are $\mathrm{T}^{\{4\}}$ and $\mathrm{T}^{\{5\}}$ but the only winning moves are $\mathrm{T}^{\{0\}}$, $\mathrm{F}^{\{1\}}$, and $\mathrm{F}^{\{2\}}$.

The detection of rational moves is in general NP-hard. The reason is that an element is dead if and only if any move in the element is rational for both players, so any polynomial-time method for detecting rational moves would also detect dead elements in polynomial time. However, as will be described in Section 13.1, detecting dead elements is NP-complete for a special case of set colouring games, and thus for set colouring games in general as well. In practice, recognizing rational moves will depend on certain special cases, such as a literal that occurs in a Boolean formula only in positive or only in negative form, or on game-specific insights.

5.3 Reversible Moves

In Combinatorial Game Theory (CGT) there is a rule that says "a reversible move can be bypassed". A move is reversible if the opponent has a reply that at least neutralizes the move. The concept of bypassing reversible moves applies to set colouring games as well.

Definition 5.3.1. Let $\Gamma = \langle X^{\mathcal{S}}, f\rangle$, $\mathfrak{p} \in \mathfrak{P}_{\mathcal{S}}$, and $\mathfrak{m} \in \mathfrak{M}(\mathfrak{p})$. If there exists $\mathfrak{m}' \in \mathfrak{M}(\mathfrak{p} \oplus \mathfrak{m})$ for which $\mathrm{NGX}(\Gamma; \mathfrak{p} \oplus \mathfrak{m} \oplus \mathfrak{m}') \leqslant \mathrm{NGX}(\Gamma; \mathfrak{p})$ then $\mathfrak{m}$ is **reversible** through $\mathfrak{m}'$.

The theorem then is that a reversible move can be replaced by all the children of the move that reverses it.

Theorem 5.3.2. Let $\Gamma = \langle X^{\mathcal{S}}, f\rangle$, $\mathfrak{p} \in \mathfrak{P}_{\mathcal{S}}$, and let $\mathfrak{m} \in \mathfrak{M}(\mathfrak{p})$ by reversible through $\mathfrak{m}' \in \mathfrak{M}(\mathfrak{p} \oplus \mathfrak{m})$. Put $\mathfrak{p}' = \mathfrak{p} \oplus \mathfrak{m}$ and $\mathfrak{p}'' = \mathfrak{p}' \oplus \mathfrak{m}'$, and $\mathfrak{P}^* = \{\xi \in \mathfrak{P}_{\mathcal{S}} \mid \xi \leftarrow \mathfrak{p} \vee \xi \leftarrow \mathfrak{p}''\}$. Then:

$$\mathrm{NGX}(\Gamma; \mathfrak{p}) = -\min_{\xi \in \mathfrak{P}^* \setminus \mathfrak{p}'} \left[\mathrm{NGX}(\Gamma; \xi)\right].$$

This theorem holds even though it is not required that $\mathfrak{m}'$ be the *best* possible reply to $\mathfrak{m}$. The reason is that $\mathfrak{m}$ was probably not optimal, in which case the reply only needed to be good enough to refute $\mathfrak{m}$, whereas if $\mathfrak{m}$ was optimal then $\mathfrak{m}'$ is retroactively guaranteed to be optimal as well in order to preserve the negamax value.

5.4 Values for Starred Games

Any game may contain dead elements, and additional dead elements may be created by considering a starred game. While dead elements do not influence the outcome function, they do have the power to influence the minimax function. The first assertion, however, is that adding a coloured dead element to the game does not make a strategic difference.

Theorem 5.4.1. Let $\Gamma = \langle X^{\mathcal{S}}, f\rangle$ and $\Gamma* = \langle X^{\mathcal{S}^*}, f\rangle$ with $\mathcal{S}^* = \mathcal{S} + w$. Let $\mathfrak{p} = (\psi, \mathfrak{c}) \in \mathfrak{P}_{\mathcal{S}}$ and $\chi \in \check{X}$. Then $\text{MNX}(\Gamma; \mathfrak{p}) = \text{MNX}(\Gamma*; \mathfrak{p}\chi^w)$.

> *Corollary i:* Let $\mathfrak{p}^* = (\psi^*, \mathfrak{c}) \in \mathfrak{P}_{\mathcal{S}^*}$ with $\psi^*_w \neq \phi$. Then $\text{MNX}(\Gamma*; \mathfrak{p}^*, \mathfrak{c}) = \text{MNX}(\Gamma; \mathfrak{p}^* \searrow \mathcal{S})$.

The main theorem refers to adding a coloured dead element, and the corollary refers to removing one; the assertion is that neither of these two mutations makes a strategic difference. The same holds for *uncoloured* dead elements, but only in pairs, as a consequence of the following theorem.

Theorem 5.4.2. Let $\Gamma = \langle X^{\mathcal{S}}, f\rangle$ and $\Gamma** = \langle X^{\mathcal{S}^{**}}, f\rangle$. Then $\text{MNX}(\Gamma; \mathfrak{p}) = \text{MNX}(\Gamma**; \mathfrak{p} \searrow \mathcal{S}^{**})$.

> *Corollary i:* Let $\mathfrak{p}^{**} = (\psi^{**}, \mathfrak{c}) \in \mathfrak{P}_{\mathcal{S}^{**}}$ with $\mathcal{S}^{**} \setminus \mathcal{S} \subseteq \mathcal{U}(\psi^{**})$. Then $\text{MNX}(\Gamma**; \mathfrak{p}^{**}) = \text{MNX}(\Gamma; \mathfrak{p}^{**} \searrow \mathcal{S})$.

This states that adding or removing two dead elements makes no strategic difference with optimal play. So any *even* number of uncoloured dead elements can effectively be ignored. The theorem does not hold for adding just one dead element, and indeed a trivial counterexample shows that adding one dead element can make a strategic difference, for if some colouring ψ is misère in some game Γ, then ψ is regular in $\Gamma*$.

From Theorem 5.4.2 it is evident that if one wants to allow a finite number of "skip moves" in a game, then one single skip move will do.

5.5 Values for Isotone Games

When the scoring function is isotone, the game and its minimax values exhibit benign properties that will be explored in this section.

Definition 5.5.1. Let $\Gamma = \langle X^{\mathcal{S}}, f\rangle$. Then Γ is isotone if and only if f is isotone.

Since the domain of f is $\check{X}^{\mathcal{S}}$, isotonicity requires $\check{X}$ to be an ordered set. This does not require X itself to be ordered; ϕ may be incomparable with some or all pure colours. Note that the players are free to choose any ordering of $\check{X}$ that may suit their tastes or purposes, since the scoring function is not affected by it, and by extension neither is the minimax function.

The important quality of an isotone game is that every element admits rational moves for both players. Conversely, any set colouring game $\Gamma = \langle X^{\mathcal{S}}, f\rangle$ that has this property is **strategically equivalent** to an isotone game. This term means that any optimal strategy for one game can be transformed into an optimal strategy for the other. To this end, define a mapping $h : \mathbb{T}^{\mathcal{S}} \to X^{\mathcal{S}}$ where each element is re-coloured with some MAX-rational colour in Γ if it was coloured T, and with some MIN-rational colour if it was coloured F. This defines an immersion from the game $\Gamma_{\mathbb{T}} = \langle \mathbb{T}^{\mathcal{S}}, f \circ h\rangle$ into Γ. The game $\Gamma_{\mathbb{T}}$ is isotone by design. Any optimal strategy in $\Gamma_{\mathbb{T}}$ is therefore mapped by h onto an optimal strategy in Γ. On the other hand, any optimal strategy for Γ can trivially be transformed into an optimal strategy for $\Gamma_{\mathbb{T}}$: whenever the Γ-strategy colours element v, then the recommendation in $\Gamma_{\mathbb{T}}$ is to colour the same element rationally.

Without loss of generality it can therefore be assumed that any isotone set colouring game is of the form $\langle \mathbb{T}^{\mathcal{S}}, f\rangle$, and, in particular, contains only two pure colours. Unfortunately the task of detecting whether an element admits rational moves is NP-hard, as a consequence of a result that will be shown in Section 13.1 to the effect that detecting dead elements is NP-hard in Hex. Detecting isotonicity of a game must therefore be NP-hard also.

If X contains only one pure colour χ then the game is trivial: it always ends with the complete colouring $\chi^{\mathcal{S}}$, and by Theorem 5.1.2:ii every position has the same minimax value $f(\chi^{\mathcal{S}})$. If χ contains two pure colours then without loss of generality $X = \mathbb{T}$, since it does not matter where ϕ appears in the ordering. This means that any isotone set colouring game with at least two pure colours is an instance of GAME-SAT.

In the remainder of this section it will be assumed that any isotone set colouring game is of the form $\langle \mathbb{T}^{\mathcal{S}}, f\rangle$. In particular this means that any legal move is of the form T^v or F^v. It can be verified readily that all the theorems also hold for trivial set colouring games with $|\check{X}| = 1$.

The first theorem is a variant of Theorems 5.4.1 and 5.4.2, which speak of adding or removing one coloured dead element and two uncoloured dead elements, respectively. In an isotone game this can be one dead element, no matter whether coloured or uncoloured.

Theorem 5.5.2. Let $\Gamma = \langle X^{\mathcal{S}}, f\rangle$ be isotone, and $\Gamma{*} = \langle X^{\mathcal{S}^*}, f\rangle$. Let $\mathfrak{p} \in \mathfrak{P}_{\mathcal{S}}$. Then $\text{MNX}(\Gamma; \mathfrak{p}) = \text{MNX}(\Gamma{*}; \mathfrak{p} \searrow \mathcal{S}^*)$.

> *Corollary i:* Let $\mathfrak{p}^* \in \mathfrak{P}_{\mathcal{S}^*}$, then $\text{MNX}(\Gamma{*}; \mathfrak{p}^*) = \text{MNX}(\Gamma; \mathfrak{p}^* \searrow \mathcal{S})$.
>
> *Corollary ii:* If $w \in \mathcal{S}$ is dead in f then $\text{MNX}(\Gamma; \mathfrak{p}) = \text{MNX}(\Gamma; \mathfrak{p}\chi^w)$ for all $\chi \in X$.
>
> *Corollary iii:* Let $\mathcal{S}' \subseteq \mathcal{S}$ such that every element $v \in \mathcal{S} \backslash \mathcal{S}'$ is dead, and let $f' = f \circ \text{PROJ}_{\mathcal{S}' \to \mathcal{S}} : X^{\mathcal{S}'} \to \mathbb{B}$. Then $\text{MNX}(\Gamma; \mathfrak{p}) = \text{MNX}\big(f'; \mathfrak{p} \searrow \mathcal{S}'\big)$.

The next theorem is similar to Theorem 5.2.2, the difference being that when the game is isotone the colours χ_- and χ_+ do not need to be *pure* colours for the theorem to apply.

Theorem 5.5.3. Let $\Gamma = \langle \mathbb{T}^{\mathcal{S}}, f\rangle$ be isotone. Choose any ordering of $\check{X}$ that satisfies $\text{T} > \phi > \text{F}$. Let $\mathfrak{p} \in \mathfrak{P}_{\mathcal{S}}$, $v \in \mathcal{S}$, and $\chi_-, \chi_+ \in X$ with $\chi_- \leqslant \chi_+$. Then $\text{MNX}(\Gamma; \mathfrak{p}\chi_-^v) \leqslant \text{MNX}(\Gamma; \mathfrak{p}\chi_+^v)$.

> *Corollary i:* $\text{MNX}(\Gamma; \mathfrak{p}\chi_-^{\mathcal{S}'}) \leqslant \text{MNX}(\Gamma; \mathfrak{p}\chi_+^{\mathcal{S}'})$ for any $\mathcal{S}' \subseteq \mathcal{S}$.

Corollary ii: $\forall_{\mathfrak{c}\in\mathfrak{C}}\forall_{\psi,\psi'\in X^S}\left[\psi \geqslant \psi' \Longrightarrow \text{MNX}(\Gamma;\psi,\mathfrak{c}) \geqslant \text{MNX}(\Gamma;\psi',\mathfrak{c})\right]$.

Corollary iii: $\text{MNX}(\Gamma;\psi,\text{MAX}) \geqslant \text{MNX}(\Gamma;\psi,\text{MIN})$.

Informally, this theorem and Corollaries i and ii state that increasing the colours of any number of elements cannot hurt MAX, and decreasing their colours cannot hurt MIN. In particular this implies that any rational move is better than skipping a move, and skipping a move is better than any irrational move. Corollary iii says that there are no misère colourings, and therefore no "zugzwang" positions.[1]

These assertions do not hold in general for monotone elements in non-isotone functions. Consider $\langle \mathbb{T}^3, f\rangle$ with $f : \xi \mapsto \xi_0 \wedge (\xi_1 \equiv \xi_2)$. Then f is increasing in element 0, and $\text{MNX}\big(f;(+,\phi,\phi),\text{MAX}\big) = \text{MNX}\big(f;(-,\phi,\phi),\text{MAX}\big) = -1$, yet $\text{MNX}\big(f;(\phi,\phi,\phi),\text{MAX}\big) = +1$.

5.6 Values under Homomorphisms

Homomorphisms provide links between games with the same structure, and automorphisms describe symmetries of a given game. When such relations in structure exist, the minimax values are also related.

Theorem 5.6.1. Let $h : X^S \to X^S$ be a pseudo-homomorphism between $\Gamma = \langle X^S, f\rangle$ and $\Gamma' = \langle X^S, f'\rangle$, with $f = t \cdot f' \circ h$ for $t \in \mathbb{B}$. If h is surjective and generation preserving, then

$$\text{MNX}\big(\Gamma'; h(\mathfrak{p})\big) = t \cdot \text{MNX}\big(\Gamma; t\cdot\mathfrak{p}\big),$$
$$\text{NGX}\big(\Gamma'; h(\mathfrak{p})\big) = \text{NGX}\big(\Gamma; t\cdot\mathfrak{p}\big)$$

for all $\mathfrak{p} \in \mathfrak{P}_S$.

Corollary i: Let $\Gamma = \langle X^S, f\rangle$ and define $-\Gamma = \langle X^S, -f\rangle$. Then $\text{MNX}(-\Gamma;\mathfrak{p}) = -\text{MNX}(\Gamma;-\mathfrak{p})$ and $\text{NGX}(-\Gamma;\mathfrak{p}) = \text{NGX}(\Gamma;-\mathfrak{p})$.

This may be abbreviated as $\pm\text{MNX}(f;\mathfrak{p}) = \text{MNX}\big(f';\pm h(\mathfrak{p})\big)$, where $\pm$ consistently takes the $+$ sign if h is an isomorphism and the $-$ sign if h is an anti-isomorphism.

Theorem 5.6.2. Let $h : X^S \to X^S$ be a generation preserving anti-automorphism of $\Gamma = \langle X^S, f\rangle$, and let $\mathfrak{p} \in \mathfrak{P}_S$ such that $h(\mathfrak{p}) = \mathfrak{p}$. Then $\text{MNX}(\Gamma;\mathfrak{p}) = -\text{MNX}(\Gamma;-\mathfrak{p})$ and $\text{NGX}(\Gamma;\mathfrak{p}) = \text{NGX}(\Gamma;-\mathfrak{p})$.

Corollary i: If furthermore f is isotone then $\text{NGX}(\Gamma;\mathfrak{p}) = +1$.

When applied to Hex, the corollary in this theorem states the fact that the game is a first player win.

[1] The term **zugzwang**, meaning "forced to move", is used in chess to indicate a position in which the player to move would prefer to skip a move.

The requirement of being generation preserving is crucial for these theorems. Pseudo-isomorphisms that occur "in practice" will typically meet this requirement, but pseudo-immersions or pseudo-contractions often do not. For instance, the theorems about adding starred games concern immersions and contractions that are not generation preserving. In the "natural" contraction from $\Gamma*$ to Γ a move in the added element of $\Gamma*$ corresponds to a pass in Γ, which is not a legal move.

5.7 Proofs

Theorem 5.1.2. Let $\langle X^{\mathcal{S}}, f\rangle$ and $\langle X^{\mathcal{S}}, f'\rangle$ be two set colouring games, and let $\mathfrak{p} = (\psi, \mathfrak{c}) \in \mathfrak{P}_{\mathcal{S}}$. Then:

i. If $\forall_{\psi^* \trianglerighteq \psi}\big[f(\psi^*) \geqslant f'(\psi^*)\big]$ then $\textsc{mnx}(f;\mathfrak{p}) \geqslant \textsc{mnx}(f';\mathfrak{p})$;

ii. If $\forall_{\psi^* \trianglerighteq \psi}\big[f(\psi) \geqslant t\big]$ for some $t \in \mathbb{B}$, then $\textsc{mnx}(f;\mathfrak{p}) \geqslant t$.

These assertions of course also hold with $\leqslant$ or $=$ substituted for $\geqslant$.

Proof. The properties are easily verified by induction to $|\mathfrak{M}(\mathfrak{p})|$. □

Theorem 5.2.2. Let $\Gamma = \langle X^{\mathcal{S}}, f\rangle$, $\mathfrak{p} \in \mathfrak{P}_{\mathcal{S}}$, and $v \in \mathcal{S}$. Let $\mathfrak{m}^-, \mathfrak{m}^+ \in \mathfrak{M}(v)$ with $\mathfrak{m}^+$ preferable to $\mathfrak{m}^-$ for MAX. Then $\textsc{mnx}(f;\mathfrak{p}\mathfrak{m}^+) \geqslant \textsc{mnx}(f;\mathfrak{p}\mathfrak{m}^-)$.

Corollary i: If $v \in \mathcal{U}(\mathfrak{p})$ then $\textsc{mnx}(f;\mathfrak{p} \oplus \mathfrak{m}^+) \geqslant \textsc{mnx}(f;\mathfrak{p} \oplus \mathfrak{m}^-)$.

Corollary ii: If v is dead in f then $\textsc{mnx}(f;\mathfrak{p}\mathfrak{m}')$ is constant for every $\mathfrak{m}' \in \mathfrak{M}(v)$.

Proof. Induction to $|\mathcal{U}(\mathfrak{p})|$.

Base case: $|\mathcal{U}(\mathfrak{p})| = 0$. Then by Definition 5.1.1 $\textsc{mnx}(f;\mathfrak{p}\mathfrak{m}^+) = f(\mathfrak{p}\mathfrak{m}^+)$ and $\textsc{mnx}(f;\mathfrak{p}\mathfrak{m}^-) = f(\mathfrak{p}\mathfrak{m}^-)$ since $\mathfrak{p}\mathfrak{m}^+$ and $\mathfrak{p}\mathfrak{m}^-$ are final positions. Then $f(\mathfrak{p}\mathfrak{m}^+) \geqslant f(\mathfrak{p}\mathfrak{m}^-)$ from Definition 5.2.1.

Induction step. If $v \in \mathcal{U}(\mathfrak{p})$ then the assertion is true by induction since $|\mathcal{U}(\mathfrak{p}\mathfrak{m}')| = |\mathcal{U}(\mathfrak{p})| - 1$ for any $\mathfrak{m} \in \mathfrak{M}(v)$ by Observation 3.2.1:iii. Assume $v \notin \mathcal{U}(\mathfrak{p})$. Then if $\mathfrak{c} = $ MAX we have

$$\begin{aligned}
\textsc{mnx}(f;\mathfrak{p}\mathfrak{m}^+) &= \max_{\mathfrak{m}' \in \mathfrak{M}(\mathfrak{p}\mathfrak{m}^+)}\big[\textsc{mnx}(f;\mathfrak{p}\mathfrak{m}^+ \oplus \mathfrak{m}')\big] && \text{(Definition 5.1.1)}\\
&= \max_{\mathfrak{m}' \in \mathfrak{M}(\mathfrak{p}\mathfrak{m}^+)}\big[\textsc{mnx}(f;(\mathfrak{p} \oplus \mathfrak{m}')\mathfrak{m}^+)\big] && \text{(Observations 3.2.2:ii and 3.2.4:i)}\\
&\geqslant \max_{\mathfrak{m}' \in \mathfrak{M}(\mathfrak{p}\mathfrak{m}^-)}\big[\textsc{mnx}(f;(\mathfrak{p} \oplus \mathfrak{m}')\mathfrak{m}^-)\big] && \text{(induction, } |\mathcal{U}(\mathfrak{p} \oplus \mathfrak{m})| < |\mathcal{U}(\mathfrak{p})|)\\
&= \max_{\mathfrak{m}' \in \mathfrak{M}(\mathfrak{p}\mathfrak{m}^-)}\big[\textsc{mnx}(f;\mathfrak{p}\mathfrak{m}^- \oplus \mathfrak{m}')\big] && \text{(Observations 3.2.2:ii and 3.2.4:i)}\\
&= \textsc{mnx}(f;\mathfrak{p}\mathfrak{m}^-) && \text{(Definition 5.1.1)}
\end{aligned}$$

and analogously if $\mathfrak{c} = $ MIN we have

$$\textsc{mnx}(f;\mathfrak{p}\mathfrak{m}^+) = \min_{\mathfrak{m}' \in \mathfrak{M}(\mathfrak{p}\mathfrak{m}^+)}[\textsc{mnx}(f;\mathfrak{p}\mathfrak{m}^+ \oplus \mathfrak{m}')] \geqslant \min_{\mathfrak{m}' \in \mathfrak{M}(\mathfrak{p}\mathfrak{m}^-)}[\textsc{mnx}(f;\mathfrak{p}\mathfrak{m}^- \oplus \mathfrak{m}')] = \textsc{mnx}(f;\mathfrak{p}\mathfrak{m}^-).$$

Corollary i follows by taking some arbitrary $\mathfrak{p}' \leftarrow \mathfrak{p}$, noting that $\mathfrak{p}'\mathfrak{m}^+ = \mathfrak{p} \oplus \mathfrak{m}^+$ and $\mathfrak{p}'\mathfrak{m}^- = \mathfrak{p} \oplus \mathfrak{m}^-$, and applying the main theorem. Corollary ii follows immediately since a function is by definition both increasing and decreasing in a dead element. □

Theorem 5.2.4. Let $\Gamma = \langle X^{\mathcal{S}}, f\rangle$, $\mathfrak{p} = (\psi, \mathfrak{c}) \in \mathfrak{P}_{\mathcal{S}}$, and $\mathfrak{m}^+, \mathfrak{m}^- \in \mathfrak{M}(\mathcal{S})$. If $\mathfrak{m}^+$ is rational for MAX and $\mathfrak{m}^-$ is rational for MIN then $\text{MNX}(f; \mathfrak{p}\mathfrak{m}^+) \geqslant \text{MNX}(f; \mathfrak{p}\mathfrak{m}^-)$.

Corollary i: $\text{MNX}(f; \mathfrak{p} \oplus \mathfrak{m}^+) \geqslant \text{MNX}(f; \mathfrak{p} \oplus \mathfrak{m}^-)$.

Proof. Put $\mathfrak{m}^+ \in \mathfrak{M}(v)$ and $\mathfrak{m}^- \in \mathfrak{M}(w)$. If $v = w$ then Theorem 5.2.2 applies, so only the case $v \neq w$ needs to be considered. The proof uses induction to $|\mathcal{U}(\mathfrak{p})|$; note that $|\mathcal{U}(\mathfrak{p})| \geqslant 2$ since $v, w \in \mathcal{U}(\mathfrak{p})$.

Base case: $|\mathcal{U}(\mathfrak{p})| = 2$. If $\mathfrak{c} = \text{MAX}$ then:

$$\begin{aligned}
\text{MNX}(f; \psi\mathfrak{m}^+, \text{MAX}) &= \max_{\mathfrak{m} \in \mathfrak{M}(\psi\mathfrak{m}^+)} \Big[\text{MNX}\big(f; (\psi\mathfrak{m}^+, \text{MAX}) \oplus \mathfrak{m}\big)\Big] && \text{(Definition 5.1.1)} \\
&= \max_{\chi \in \check{X}} \Big[\text{MNX}\big(f; (\psi\mathfrak{m}^+, \text{MAX}) \oplus \chi^w\big)\Big] && \text{(Definition 3.2.2, } \mathcal{U}(\psi\mathfrak{m}^+) = \{w\}) \\
&= \max_{\chi \in \check{X}} \Big[\text{MNX}\big(f; \psi\mathfrak{m}^+\chi^w, \text{MIN}\big)\Big] && \text{(Definition 3.2.4)} \\
&= \max_{\chi \in \check{X}} \Big[f\big(\psi\mathfrak{m}^+\chi^w\big)\Big] && \text{(Definition 5.1.1, } \psi\mathfrak{m}^+\chi^w \text{ is final)} \\
&\geqslant f\big(\psi\mathfrak{m}^+\mathfrak{m}^-\big) && (\mathfrak{m}^- \text{ is rational for MIN})
\end{aligned}$$

and similarly $\text{MNX}(f; \psi\mathfrak{m}^-) = \max_{\chi \in \check{X}} [f(\psi\mathfrak{m}^-\chi^v)] = f(\psi\mathfrak{m}^-\mathfrak{m}^+)$ since $\mathfrak{m}^+$ is rational for MAX. Therefore $\text{MNX}(f; \psi\mathfrak{m}^+) \geqslant f(\psi\mathfrak{m}^+\mathfrak{m}^-) = f(\psi\mathfrak{m}^-\mathfrak{m}^+) = \text{MNX}(f; \psi\mathfrak{m}^-)$. Analogously, if $\mathfrak{c} = \text{MIN}$ then

$$\text{MNX}(f; \psi\mathfrak{m}^-, \text{MIN}) = \min_{\chi \in \check{X}} \big[f(\psi\mathfrak{m}^-\chi^v)\big] \leqslant f(\psi\mathfrak{m}^-\mathfrak{m}^+) = f(\psi\mathfrak{m}^+\mathfrak{m}^-) = \text{MNX}(f; \psi\mathfrak{m}^+, \text{MIN}).$$

Induction step. Put $\mathcal{S}' = \mathcal{U}(\mathfrak{p}) - v - w$. Since $\mathcal{U}(\mathfrak{p}\mathfrak{m}^-) = \mathcal{U}(\mathfrak{p}) - w$, any child of $\mathfrak{p}\mathfrak{m}^+$ is of the form $(\psi\mathfrak{m}^+\chi^w, \bar{\mathfrak{c}})$ or $(\psi\mathfrak{m}^+\chi^u, \bar{\mathfrak{c}})$, and any child of $\mathfrak{p}\mathfrak{m}^-$ is of the form $(\psi\mathfrak{m}^-\chi^v, \bar{\mathfrak{c}})$ or $(\psi\mathfrak{m}^-\chi^u, \bar{\mathfrak{c}})$, for some $\chi \in \check{X}$ and $u \in \mathcal{S}'$. We then have:

$$\begin{aligned}
\text{MNX}(f; \psi\mathfrak{m}^+\chi^w, \bar{\mathfrak{c}}) &\geqslant \text{MNX}(f; \psi\mathfrak{m}^-\mathfrak{m}^+, \bar{\mathfrak{c}}) \\
\text{MNX}(f; \psi\mathfrak{m}^+\mathfrak{m}^-, \bar{\mathfrak{c}}) &\geqslant \text{MNX}(f; \psi\mathfrak{m}^-\chi^v, \bar{\mathfrak{c}}) \\
\text{MNX}(f; \psi\mathfrak{m}^+\chi^u, \bar{\mathfrak{c}}) &\geqslant \text{MNX}(f; \psi\mathfrak{m}^-\chi^u, \bar{\mathfrak{c}})
\end{aligned}$$

Since every child of $\mathfrak{p}\mathfrak{m}^+$ occurs at least once in the left hand column, we have

$$\text{MNX}(f; \psi\mathfrak{m}^+, \text{MAX}) = \max_{\xi \leftarrow \mathfrak{p}\mathfrak{m}^+} \big[\text{MNX}(f; \xi)\big] \geqslant \max_{\xi \leftarrow \mathfrak{p}\mathfrak{m}^-} \big[\text{MNX}(f; \xi)\big] = \text{MNX}(f; \psi\mathfrak{m}^-, \text{MAX}).$$

Similarly, since every child of $\mathfrak{p}\mathfrak{m}^-$ occurs at least once in the right hand column,

$$\text{MNX}(f; \psi\mathfrak{m}^-, \text{MIN}) = \min_{\xi \leftarrow \mathfrak{p}\mathfrak{m}^-} \big[\text{MNX}(f; \xi)\big] \leqslant \min_{\xi \leftarrow \mathfrak{p}\mathfrak{m}^+} \big[\text{MNX}(f; \xi)\big] = \text{MNX}(f; \psi\mathfrak{m}^+, \text{MIN})$$

which proves the theorem. □

Theorem 5.2.5. Let $\Gamma = \langle X^{\mathcal{S}}, f\rangle$ and $\mathfrak{p} = (\psi, \mathfrak{c}) \in \mathfrak{P}_{\mathcal{S}}$. Let $\mathfrak{m}, \chi^w \in \mathfrak{M}(\mathfrak{p})$ where w is dead in f. If $\mathfrak{m}$ is rational for $\mathfrak{c}$ then $\text{NGX}(f;\mathfrak{p} \oplus \mathfrak{m}) \leqslant \text{NGX}(f;\mathfrak{p} \oplus \chi^w)$. If $\mathfrak{m}$ is irrational for $\mathfrak{c}$ then $\text{NGX}(f;\mathfrak{p} \oplus \mathfrak{m}) \geqslant \text{NGX}(f;\mathfrak{p} \oplus \chi^w)$

Proof. Since χ^w is rational and irrational for both MIN and MAX by Observation 5.2.3:i, the theorem then follows directly by applying Theorem 5.2.4. □

Theorem 5.3.2. Let $\Gamma = \langle X^{\mathcal{S}}, f\rangle$, $\mathfrak{p} \in \mathfrak{P}_{\mathcal{S}}$, and let $\mathfrak{m} \in \mathfrak{M}(\mathfrak{p})$ by reversible through $\mathfrak{m}' \in \mathfrak{M}(\mathfrak{p} \oplus \mathfrak{m})$. Put $\mathfrak{p}' = \mathfrak{p} \oplus \mathfrak{m}$ and $\mathfrak{p}'' = \mathfrak{p}' \oplus \mathfrak{m}'$, and $\mathfrak{P}^* = \{\xi \in \mathfrak{P}_{\mathcal{S}} \mid \xi \leftarrow \mathfrak{p} \vee \xi \leftarrow \mathfrak{p}''\}$. Then:

$$\text{NGX}(\Gamma;\mathfrak{p}) = -\min_{\xi \in \mathfrak{P}^* \setminus \mathfrak{p}'} \left[\text{NGX}(\Gamma;\xi)\right].$$

Proof. The equation holds if and only if the following two conditions are satisfied:

- $\forall_{\xi \in \mathfrak{P}^* \setminus \mathfrak{p}'}[\text{NGX}(\Gamma;\xi) \geqslant -\text{NGX}(\Gamma;\mathfrak{p})]$;
- $\exists_{\xi \in \mathfrak{P}^* \setminus \mathfrak{p}'}[\text{NGX}(\Gamma;\xi) = -\text{NGX}(\Gamma;\mathfrak{p})]$.

If $\xi \in \mathfrak{P}^* \setminus \mathfrak{p}'$ then $\xi \leftarrow \mathfrak{p}$ or $\xi \leftarrow \mathfrak{p}''$. When $\xi \leftarrow \mathfrak{p}$ then $\text{NGX}(\Gamma;\xi) \geqslant -\text{NGX}(\Gamma;\mathfrak{p})$ by Observation 5.1.3:iv, and if $\xi \leftarrow \mathfrak{p}''$ then $\text{NGX}(\Gamma;\xi) \geqslant -\text{NGX}(\Gamma;\mathfrak{p}'') \geqslant -\text{NGX}(\Gamma;\mathfrak{p})$ since $\text{NGX}(\Gamma;\mathfrak{p}'') \leqslant \text{NGX}(\Gamma;\mathfrak{p})$ by Definition 5.3.1. To prove that there exists $\xi \in \mathfrak{P}^* \setminus \mathfrak{p}'$ such that $\text{NGX}(\Gamma;\xi) = -\text{NGX}(\Gamma;\mathfrak{p})$, distinguish two cases: $\text{NGX}(\Gamma;\mathfrak{p}') = -\text{NGX}(\Gamma;\mathfrak{p})$ and $\text{NGX}(\Gamma;\mathfrak{p}') \neq -\text{NGX}(\Gamma;\mathfrak{p})$.

Case: $\text{NGX}(\Gamma;\mathfrak{p}') = -\text{NGX}(\Gamma;\mathfrak{p})$. Then $\text{NGX}(\Gamma;\mathfrak{p}'') \geqslant -\text{NGX}(\Gamma;\mathfrak{p}') = \text{NGX}(\Gamma;\mathfrak{p})$. Since $\mathfrak{m}$ was reversible we also have $\text{NGX}(\Gamma;\mathfrak{p}'') \leqslant \text{NGX}(\Gamma;\mathfrak{p})$, and therefore $\text{NGX}(\Gamma;\mathfrak{p}'') = \text{NGX}(\Gamma;\mathfrak{p})$. By Definition 5.1.3 there then exists $\xi \leftarrow \mathfrak{p}''$ such that $\text{NGX}(\Gamma;\xi) = -\text{NGX}(\Gamma;\mathfrak{p}'') = -\text{NGX}(\Gamma;\mathfrak{p})$. For this ξ we have $\xi \in \mathfrak{P}^*$, and $\xi \neq \mathfrak{p}'$ because $\xi \leftarrow \mathfrak{p}'' \leftarrow \mathfrak{p}'$. Therefore $\xi \in \mathfrak{P}^* \setminus \mathfrak{p}'$.

Case: $\text{NGX}(\Gamma;\mathfrak{p}') \neq -\text{NGX}(\Gamma;\mathfrak{p})$. From Definition 5.1.3 there exists $\xi \leftarrow \mathfrak{p}$ such that $\text{NGX}(\Gamma;\xi) = -\text{NGX}(\Gamma;\mathfrak{p})$. As $\text{NGX}(\Gamma;\mathfrak{p}') \neq -\text{NGX}(\Gamma;\mathfrak{p})$ we have that $\xi \neq \mathfrak{p}'$, and therefore $\xi \in \mathfrak{P}^* \setminus \mathfrak{p}'$. □

Theorem 5.4.1. Let $\Gamma = \langle X^{\mathcal{S}}, f\rangle$ and $\Gamma* = \langle X^{\mathcal{S}^*}, f\rangle$ with $\mathcal{S}^* = \mathcal{S} + w$. Let $\mathfrak{p} = (\psi, \mathfrak{c}) \in \mathfrak{P}_{\mathcal{S}}$ and $\chi \in \check{X}$. Then $\text{MNX}(\Gamma;\mathfrak{p}) = \text{MNX}(\Gamma*;\mathfrak{p}\chi^w)$.

Corollary i: Let $\mathfrak{p}^* = (\psi^*, \mathfrak{c}) \in \mathfrak{P}_{\mathcal{S}^*}$ with $\psi^*_w \neq \phi$. Then $\text{MNX}(\Gamma*;\mathfrak{p}^*, \mathfrak{c}) = \text{MNX}(\Gamma;\mathfrak{p}^* \searrow \mathcal{S})$.

Proof. Induction to $|\mathcal{U}(\mathfrak{p})|$. The theorem is equivalent to $\text{NGX}(f;\mathfrak{p}) = \text{NGX}(f^*;\mathfrak{p}^*)$ with $f^* = f \circ \text{PROJ}_{\mathcal{S}^* \to \mathcal{S}}$. Note that $\mathcal{U}(\mathfrak{p}) = \mathcal{U}(\mathfrak{p}^*)$ by Observations 3.2.1:iii and 3.2.1:iv, and therefore also $\mathfrak{M}(\mathfrak{p}) = \mathfrak{M}(\mathfrak{p}^*)$.

Base case: $|\mathcal{U}(\mathfrak{p})| = 0$.

$$\begin{aligned}
\text{MNX}(\Gamma*;\mathfrak{p}^*) &= f^*\big((\psi \searrow \mathcal{S}^*)\chi^w\big) && \text{(Definition 5.1.1)}\\
&= f\big((\psi \searrow \mathcal{S}^*)\chi^w \searrow \mathcal{S}\big) && \text{(Definition 3.1.5)}\\
&= f\big(\psi \searrow \mathcal{S}^* \searrow \mathcal{S}\big) && \text{(Lemma 2.2.2:iii, } \chi^w \in X^{\mathcal{S}^* \setminus \mathcal{S}})\\
&= f(\psi) && \text{(Lemma 2.2.2:v, } \mathcal{S} \setminus \mathcal{S}^* = \varnothing)\\
&= \text{MNX}(\Gamma;\mathfrak{p}) && \text{(Definition 5.1.1).}
\end{aligned}$$

Induction step: $|\mathcal{U}(\mathfrak{p})| > 0$.

$$\begin{aligned}
\text{NGX}(\Gamma*;\mathfrak{p}^*) &= -\min_{\mathfrak{m}\in\mathfrak{M}(\mathfrak{p}^*)}\big[\text{NGX}(\Gamma*;\mathfrak{p}^*\oplus\mathfrak{m})\big] && \text{(Observation 5.1.3:i)}\\
&= -\min_{\mathfrak{m}\in\mathfrak{M}(\mathfrak{p})}\big[\text{NGX}(\Gamma;\mathfrak{p}\oplus\mathfrak{m})\big] && \text{(induction, } \mathfrak{M}(\mathfrak{p}) = \mathfrak{M}(\mathfrak{p}^*))\\
&= \text{NGX}(\Gamma;\mathfrak{p}) && \text{(Observation 5.1.3:i).}
\end{aligned}$$

which proves the main theorem. □

Proof of corollary. Choose $\psi = \psi^* \searrow \mathcal{S}$ and $\chi = \psi^*_w$, then $\mathfrak{p}^* = \mathfrak{p}\chi^w$ and $\chi \in \check{X}$, so the main theorem applies. □

Theorem 5.4.2. Let $\Gamma = \langle X^{\mathcal{S}}, f\rangle$ and $\Gamma{*}{*} = \langle X^{\mathcal{S}^{**}}, f\rangle$. Then $\text{MNX}(\Gamma;\mathfrak{p}) = \text{MNX}(\Gamma{*}{*};\mathfrak{p} \searrow \mathcal{S}^{**})$.

> *Corollary i:* Let $\mathfrak{p}^{**} = (\psi^{**}, \mathfrak{c}) \in \mathfrak{P}_{\mathcal{S}^{**}}$ with $\mathcal{S}^{**} \setminus \mathcal{S} \subseteq \mathcal{U}(\psi^{**})$. Then $\text{MNX}(\Gamma**;\mathfrak{p}^{**}) = \text{MNX}(\Gamma;\mathfrak{p}^{**} \searrow \mathcal{S})$.

Proof. Put $\mathfrak{p}^{**} = \mathfrak{p} \searrow \mathcal{S}^{**}$. The theorem is equivalent to $\text{NGX}(\Gamma;\mathfrak{p}) = \text{NGX}(\Gamma**;\mathfrak{p}^{**})$. If $\mathfrak{M}(\mathfrak{p}) = \varnothing$ then the theorem follows from Theorem 5.1.2:ii since $f(\psi') = f(\psi)$ for all $\psi' \trianglerighteq \psi \searrow \mathcal{S}^{**}$, so for the rest of the proof it may be assumed that $\mathfrak{M}(\mathfrak{p})$ is not empty. Put $\mathfrak{M}^{**} = \{\chi^w | \chi \in \check{X}, w \in \mathcal{S}^{**} \setminus \mathcal{S}\}$, then by Definition 3.2.2 we have $\mathfrak{M}(\mathfrak{p}^{**}) = \mathfrak{M}(\mathfrak{p}) \cup \mathfrak{M}^{**}$. First,

$$\text{NGX}(\Gamma;\mathfrak{p}) = -1 \Longrightarrow \text{NGX}(\Gamma**;\mathfrak{p}^{**}) = -1 \tag{5.7.1}$$

by induction on $|\mathfrak{M}(\mathfrak{p})|$. Let $\text{NGX}(\Gamma;\mathfrak{p}) = -1$. If $|\mathfrak{M}(\mathfrak{p})| = 0$ then $\mathfrak{M}(\mathfrak{p}) = \varnothing$ in which case $\text{NGX}(\Gamma**;\mathfrak{p}^{**}) = \text{NGX}(\Gamma;\mathfrak{p})$ as mentioned above. If $|\mathfrak{M}(\mathfrak{p})| > 0$ then, by the observations from Definition 5.1.3, it is sufficient to prove that $\forall_{\mathfrak{m}\in\mathfrak{M}(\mathfrak{p}^{**})}\big[\text{NGX}(\Gamma**;\mathfrak{p}^{**}\oplus\mathfrak{m}) = +1\big]$. Let $\mathfrak{m} \in \mathfrak{M}(\mathfrak{p}^{**})$, then there are two cases:

- $\mathfrak{m} \in \mathfrak{M}(\mathfrak{p})$. Then $\text{NGX}(\Gamma;\mathfrak{p}\oplus\mathfrak{m}) = +1$, so there exists $\mathfrak{m}' \in \mathfrak{M}(\mathfrak{p}\oplus\mathfrak{m})$ with $\text{NGX}(\Gamma;\mathfrak{p}\oplus\mathfrak{m}\oplus\mathfrak{m}') = -1$. Then $\text{NGX}(\Gamma**;\mathfrak{p}^{**}\oplus\mathfrak{m}\oplus\mathfrak{m}') = -1$ by the induction hypothesis, and therefore $\text{NGX}(\Gamma**;\mathfrak{p}^{**}\oplus\mathfrak{m}) = +1$.
- $\mathfrak{m} \notin \mathfrak{M}(\mathfrak{p})$. Then $\mathfrak{m} = \chi^w$ for some $w \in \mathcal{S}^{**} \setminus \mathcal{S}$ and $\chi \in \check{X}$. Let $\mathfrak{m}' = \chi^{w'}$ for $w' = \mathcal{S}^{**} \setminus \mathcal{S} - w$. Then $\text{NGX}(\Gamma**;\mathfrak{p}^{**}\oplus\mathfrak{m}\oplus\mathfrak{m}') = \text{NGX}(\Gamma;\mathfrak{p})$ by Lemma 5.4.1 because w and w' are dead in f^{**}. Therefore $\text{NGX}(\Gamma**;\mathfrak{p}^{**}\oplus\mathfrak{m}) = +1$.

In both cases we have $\text{NGX}(\Gamma**;\mathfrak{p}^{**}\oplus\mathfrak{m}) = +1$ which proves Implication 5.7.1. Next,

$$\text{NGX}(\Gamma;\mathfrak{p}) = +1 \Longrightarrow \text{NGX}(\Gamma**;\mathfrak{p}^{**}) = +1. \tag{5.7.2}$$

If $\text{NGX}(\Gamma;\mathfrak{p}) = +1$ then there exists $\mathfrak{m}^+ \in \mathfrak{M}(\mathfrak{p})$ such that $\text{NGX}(\Gamma;\mathfrak{p}\oplus\mathfrak{m}^+) = -1$. Note that $\mathfrak{m}^+ \in \mathfrak{M}(\mathfrak{p}^{**})$ since $\mathfrak{M}(\mathfrak{p}) \subseteq \mathfrak{M}(\mathfrak{p}^{**})$. Then from Implication 5.7.1 we have $\text{NGX}(\Gamma;\Gamma**)\mathfrak{p}^{**}\oplus\mathfrak{m}^+ = -1$ and therefore $\text{NGX}(\Gamma**;\mathfrak{p}) = +1$. The main theorem now follows from combining implications 5.7.1 and 5.7.2. □

Note that the proof works because, crucially, in the case $\mathfrak{m} \notin \mathfrak{M}(\mathfrak{p})$ for Implication 5.7.1 the move $\mathfrak{m}'$ is guaranteed to exists since $\mathcal{S}^{**}\setminus\mathcal{S}$ contains two elements. The proof would not work for adding just one dead element.

Proof of corollary. Choose $\psi = \psi^{**} \searrow \mathcal{S}$, then $\mathfrak{p}^{**} = (\psi \searrow \mathcal{S}^{**}, \mathfrak{c})$, so the main theorem applies. □

Theorem 5.5.2. Let $\Gamma = \langle X^{\mathcal{S}}, f\rangle$ be isotone, and $\Gamma* = \langle X^{\mathcal{S}^*}, f\rangle$. Let $\mathfrak{p} \in \mathfrak{P}_{\mathcal{S}}$. Then $\text{MNX}(\Gamma;\mathfrak{p}) = \text{MNX}(\Gamma*;\mathfrak{p}\searrow\mathcal{S}^*)$.

Corollary i: Let $\mathfrak{p}^* \in \mathfrak{P}_{\mathcal{S}^*}$, then $\text{MNX}(\Gamma*;\mathfrak{p}^*) = \text{MNX}(\Gamma;\mathfrak{p}^*\searrow\mathcal{S})$.

Corollary ii: If $w \in \mathcal{S}$ is dead in f then $\text{MNX}(\Gamma;\mathfrak{p}) = \text{MNX}(\Gamma;\mathfrak{p}\chi^w)$ for all $\chi \in X$.

Corollary iii: Let $\mathcal{S}' \subseteq \mathcal{S}$ such that every element $v \in \mathcal{S}\setminus\mathcal{S}'$ is dead, and let $f' = f \circ \text{PROJ}_{\mathcal{S}'\to\mathcal{S}} : X^{\mathcal{S}'} \to \mathbb{B}$. Then $\text{MNX}(\Gamma;\mathfrak{p}) = \text{MNX}(f';\mathfrak{p}\searrow\mathcal{S}')$.

Proof of main theorem. Induction on $|\mathcal{U}(\mathfrak{p})|$. Put $\mathfrak{p}^* = \mathfrak{p}\searrow\mathcal{S}^*$.

Base case: $|\mathcal{U}(\mathfrak{p})| = 0$. In this case $\mathfrak{p}$ is a final position, so $\text{MNX}(\Gamma;\mathfrak{p}) = f(\psi)$. Let $\psi^* = \psi\searrow\mathcal{S}^*$. Since $\mathcal{U}(\psi^*) = w$, any completion of ψ^* is of the form $\psi^*\chi^w$ for some $\chi \in \check{X}$. So for any completion $\psi^*\chi^w$ of ψ^* we have

$$\begin{aligned} f(\psi^*\chi^w) &= f(\psi^*\chi^w\searrow\mathcal{S}) && \text{(Definition 2.3.4)}\\ &= f((\psi\searrow\mathcal{S}^*)\chi^w\searrow\mathcal{S}) && \text{(definition of } \psi^*)\\ &= f(\psi\searrow\mathcal{S}^*\searrow\mathcal{S}) && \text{(Lemma 2.2.2:iii)}\\ &= f(\psi) && \text{(Lemma 2.2.2:v)} \end{aligned}$$

so then Theorem 5.1.2:ii implies $\text{MNX}(\Gamma*;\mathfrak{p}^*) = f(\psi) = \text{MNX}(\Gamma;\mathfrak{p})$.

Induction step: $|\mathcal{U}(\mathfrak{p})| > 0$. Distinguish the cases $\text{NGX}(\Gamma;\mathfrak{p}) = +1$ and $\text{NGX}(\Gamma;\mathfrak{p}) = -1$.

- If $\text{NGX}(\Gamma;\mathfrak{p}) = +1$ then there exists $\mathfrak{m} \in \mathfrak{M}(\mathfrak{p})$ such that $\text{NGX}(\Gamma;\mathfrak{p}\oplus\mathfrak{m}) = -1$. Since $\mathfrak{M}(\mathfrak{p}) \subseteq \mathfrak{M}(\mathfrak{p}^*)$ this means that $\text{NGX}(\Gamma;\Gamma*)\mathfrak{p}^*\oplus\mathfrak{m} = \text{NGX}(\Gamma;\mathfrak{p}\oplus\mathfrak{m}) = -1$ by induction, and so $\text{NGX}(\Gamma;\Gamma*)\mathfrak{p}^* = +1$.
- If $\text{NGX}(\Gamma;\mathfrak{p}) = -1$ then for every $\mathfrak{m} \in \mathfrak{M}(\mathfrak{p})$ we have $\text{NGX}(\Gamma;\mathfrak{p}\oplus\mathfrak{m}) = +1$. Let $\mathfrak{m}^* \in \mathfrak{M}(\mathfrak{p}^*)$. If $\mathfrak{m}^* \in \mathfrak{M}(\mathfrak{p})$ then $\text{NGX}(\Gamma*;\mathfrak{p}^*\oplus\mathfrak{m}^*) = \text{NGX}(\Gamma;\mathfrak{p}\oplus\mathfrak{m}^*) = +1$ by induction. If $\mathfrak{m}^* \notin \mathfrak{M}(\mathfrak{p})$ then $\mathfrak{m}^* = \chi^w$ for some $\chi \in \check{X}$, since $\mathcal{U}(\mathfrak{p}^*) = \mathcal{U}(\mathfrak{p}) + w$. Then according to Theorem 5.2.5 there exists $\mathfrak{m}' \in \mathfrak{M}(\mathfrak{p})$ such that $\text{NGX}(\Gamma;\Gamma*)\mathfrak{p}^*\oplus\mathfrak{m}^* \geqslant \text{NGX}(\Gamma;\mathfrak{p}^*\oplus\mathfrak{m}') = +1$.

In both cases we have $\text{MNX}(\Gamma;\mathfrak{p}) = \text{MNX}(\Gamma*;\mathfrak{p}^*)$, which proves the theorem. □

Proof of Corollary i. Put $\mathfrak{p}^* = (\psi^*, \mathfrak{c})$. If $\psi^*_w = \phi$ then $\psi^* \searrow \mathcal{S} \searrow \mathcal{S}^* = \psi^*\phi^w = \psi^*$ by Lemma 2.2.2:v, so the main theorem applies. If $\psi^*_w \neq \phi$ then Theorem 5.4.1:i applies. □

Proof of Corollary ii. Put $\mathcal{S}' = \mathcal{S} \setminus w$ and $\Gamma' = \langle X^{\mathcal{S}'}, f\rangle$. Then $\Gamma'* = \Gamma$, and Corollary i and Lemma 2.2.2:iii imply that $\text{MNX}(\Gamma; \mathfrak{p}) = \text{MNX}(\Gamma'; \mathfrak{p} \searrow \mathcal{S}') = \text{MNX}(\Gamma'; (\mathfrak{p}\chi^w) \searrow \mathcal{S}') = \text{MNX}(\Gamma; \mathfrak{p}\chi^w)$. □

Proof of Corollary iii. This follows from repeatedly applying Corollary i to each of the elements of $\mathcal{S} \setminus \mathcal{S}'$. □

Theorem 5.5.3. Let $\Gamma = \langle \mathbb{T}^{\mathcal{S}}, f\rangle$ be isotone. Choose any ordering of $\check{X}$ that satisfies $\text{T} > \phi > \text{F}$. Let $\mathfrak{p} \in \mathfrak{P}_{\mathcal{S}}$, $v \in \mathcal{S}$, and $\chi_-, \chi_+ \in X$ with $\chi_- \leqslant \chi_+$. Then $\text{MNX}(\Gamma; \mathfrak{p}\chi^v_-) \leqslant \text{MNX}(\Gamma; \mathfrak{p}\chi^v_+)$.

> *Corollary i:* $\text{MNX}(\Gamma; \mathfrak{p}\chi^{\mathcal{S}'}_-) \leqslant \text{MNX}(\Gamma; \mathfrak{p}\chi^{\mathcal{S}'}_+)$ for any $\mathcal{S}' \subseteq \mathcal{S}$.
>
> *Corollary ii:* $\forall_{\mathfrak{c}\in\mathfrak{C}} \forall_{\psi,\psi' \in X^{\mathcal{S}}} \big[\psi \geqslant \psi' \Longrightarrow \text{MNX}(\Gamma; \psi, \mathfrak{c}) \geqslant \text{MNX}(\Gamma; \psi', \mathfrak{c})\big]$.
>
> *Corollary iii:* $\text{MNX}(\Gamma; \psi, \text{MAX}) \geqslant \text{MNX}(\Gamma; \psi, \text{MIN})$.

Proof of main theorem. If $\chi_- \neq \phi$ and $\chi_+ \neq \phi$ then the theorem follows from Theorem 5.2.2. If $\chi_- = \chi_+ = \phi$ then there is nothing to prove. There are two cases left: $\text{MNX}(\Gamma; \mathfrak{p}\text{F}^v) \leqslant \text{MNX}(\Gamma; \mathfrak{p}\phi^v)$ and $\text{MNX}(\Gamma; \mathfrak{p}\phi^v) \leqslant \text{MNX}(\Gamma; \mathfrak{p}\text{T}^v)$. Consider $\Gamma* = \langle X^{\mathcal{S}^*}, f\rangle$ with $\mathcal{S}^* = \mathcal{S} + w$, so that w is dead in $\Gamma*$. Let $\psi^* = \psi\phi^v \searrow \mathcal{S}^* \in X^{\mathcal{S}^*}$. Then:

$$\begin{aligned}
\text{MNX}(\Gamma; \psi\text{F}^v, \mathfrak{c}) &= \text{MNX}(\Gamma*; \psi\text{F}^v \searrow \mathcal{S}^*, \mathfrak{c}) && \text{(Theorem 5.5.2)}\\
&= \text{MNX}(\Gamma*; \psi\phi^v\text{F}^v \searrow \mathcal{S}^*, \mathfrak{c}) && \text{(Observation 2.2.1:ii)}\\
&= \text{MNX}(\Gamma*; (\psi\phi^v \searrow \mathcal{S}^*)\text{F}^v, \mathfrak{c}) && \text{(Lemma 2.2.2:ii)}\\
&= \text{MNX}(\Gamma*; \psi^*\text{F}^v, \mathfrak{c}) && \text{(definition of } \psi^*)\\
&\leqslant \text{MNX}(\Gamma*; \psi^*\text{F}^w, \mathfrak{c}) && \text{(Theorem 5.2.5, } \psi^*\text{F}^v \leftarrow \psi^*,\ \psi^*\text{F}^w \leftarrow \psi^*)\\
&= \text{MNX}(\Gamma; \psi^*\text{F}^w \searrow \mathcal{S}, \mathfrak{c}) && \text{(Corollary 5.5.2:iii)}\\
&= \text{MNX}(\Gamma; \psi^* \searrow \mathcal{S}, \mathfrak{c}) && \text{(Lemma 2.2.2:iii)}\\
&= \text{MNX}(\Gamma; \psi\phi^v, \mathfrak{c}) && \text{(Lemma 2.2.2:v)}
\end{aligned}$$

and similarly $\text{MNX}(\Gamma; \psi\text{T}^v, \mathfrak{c}) = \text{MNX}(\Gamma*; \psi^*\text{T}^v, \mathfrak{c}) \geqslant \text{MNX}(\Gamma*; \psi^*\text{T}^w, \mathfrak{c}) = \text{MNX}(\Gamma; \psi\phi^v, \mathfrak{c})$. □

Proof of Corollaries i–ii. Corollary i follows by induction to $|\mathcal{S}'|$, and Corollary ii follows by induction on the number of elements for which $\psi_i \neq \psi'_i$. □

Proof of Corollary iii. If ψ is a complete colouring then $\text{MNX}(\Gamma; \psi, \text{MAX}) = f(\psi) = \text{MNX}(\Gamma; \psi, \text{MIN})$. If ψ is not complete, then pick an arbitrary $v \in \mathcal{U}(\psi)$. We then have

$$\begin{aligned}
\text{MNX}(\Gamma; \psi, \text{MAX}) &= \max_{\mathfrak{m}\in\mathfrak{M}(\psi,\text{MAX})} \Big[\text{MNX}\big(\Gamma; (\psi, \text{MAX}) \oplus \mathfrak{m}, \text{MAX}\big)\Big] && \text{(Definition 5.1.1)}\\
&\geqslant \text{MNX}\big(\Gamma; (\psi, \text{MAX}) \oplus \text{T}^v, \text{MAX}\big) && (\text{T}^v \in \mathfrak{M}(\psi, \text{MAX}))\\
&= \text{MNX}(\Gamma; \psi\text{T}^v, \text{MIN}) && \text{(Definition 3.2.4)}\\
&\geqslant \text{MNX}(\Gamma; \psi, \text{MIN}) && \text{(main theorem)}
\end{aligned}$$

and conversely $\text{MNX}(\Gamma;\psi,\text{MIN}) \leqslant \text{MNX}(\Gamma;\psi\text{F}^v,\text{MAX}) \leqslant \text{MNX}(\Gamma;\psi,\text{MAX})$. □

Theorem 5.6.1. Let $h : X^{\mathcal{S}} \to X^{\mathcal{S}}$ be a pseudo-homomorphism between $\Gamma = \langle X^{\mathcal{S}}, f\rangle$ and $\Gamma' = \langle X^{\mathcal{S}}, f'\rangle$, with $f = t \cdot f' \circ h$ for $t \in \mathbb{B}$. If h is surjective and generation preserving, then

$$\begin{aligned}\text{MNX}\big(\Gamma'; h(\mathfrak{p})\big) &= t \cdot \text{MNX}\big(\Gamma; t\cdot\mathfrak{p}\big),\\ \text{NGX}\big(\Gamma'; h(\mathfrak{p})\big) &= \text{NGX}\big(\Gamma; t\cdot\mathfrak{p}\big)\end{aligned}$$

for all $\mathfrak{p} \in \mathfrak{P}_{\mathcal{S}}$.

Corollary i: Let $\Gamma = \langle X^{\mathcal{S}}, f\rangle$ and define $-\Gamma = \langle X^{\mathcal{S}}, -f\rangle$. Then $\text{MNX}(-\Gamma;\mathfrak{p}) = -\text{MNX}(\Gamma;-\mathfrak{p})$ and $\text{NGX}(-\Gamma;\mathfrak{p}) = \text{NGX}(\Gamma;-\mathfrak{p})$.

Proof. For any position $(\psi,\mathfrak{c})$ we have $t\cdot h(\psi,\mathfrak{c}) = (h(\psi), t\cdot\mathfrak{c}) = h(t\cdot(\psi,\mathfrak{c}))$ from Definitions 3.2.3 and 3.4.1. The two equations are equivalent, for

$$\begin{aligned}\text{MNX}(\Gamma'; h(\psi,\mathfrak{c})) &= t\cdot\text{MNX}(\Gamma; t\cdot(\psi,\mathfrak{c})) &&\Longleftrightarrow\\ \text{MNX}(\Gamma'; h(\psi),\mathfrak{c}) &= t\cdot\text{MNX}(\Gamma; \psi, t\cdot\mathfrak{c}) &&\Longleftrightarrow\\ \lambda(\mathfrak{c})\cdot\text{MNX}(\Gamma'; h(\psi),\mathfrak{c}) &= \lambda(\mathfrak{c})\cdot t\cdot\text{MNX}(\Gamma; \psi, t\cdot\mathfrak{c}) &&\Longleftrightarrow\\ \lambda(\mathfrak{c})\cdot\text{MNX}(\Gamma'; h(\psi),\mathfrak{c}) &= \lambda(t\cdot\mathfrak{c})\cdot\text{MNX}(\Gamma; \psi, t\cdot\mathfrak{c}) &&\Longleftrightarrow\\ \text{NGX}(\Gamma'; h(\psi),\mathfrak{c}) &= \text{NGX}(\Gamma; \psi, t\cdot\mathfrak{c}) &&\Longleftrightarrow\\ \text{NGX}(\Gamma'; h(\psi,\mathfrak{c})) &= \text{NGX}(\Gamma; t\cdot(\psi,\mathfrak{c})).\end{aligned}$$

The theorem follows from induction on $|\mathcal{U}(\mathfrak{p})|$.

Base case: $\mathcal{U}(\mathfrak{p}) = \varnothing$. If $h(\mathfrak{p})$ has any children then they are of the form $h(\mathfrak{p}')$ for $\mathfrak{p}' \in \mathfrak{P}_{\mathcal{S}}$ since h is surjective, and then $\mathfrak{p}'$ is a child of $\mathfrak{p}$ because h is generation preserving. But $\mathfrak{p}$ is a final position, so $h(\mathfrak{p})$ has no children and must be a final position as well. Let $\mathfrak{p} = (\psi,\mathfrak{c})$, then $t\cdot\text{MNX}(\Gamma; t\cdot(\psi,\mathfrak{c})) = t\cdot\text{MNX}(\Gamma;\psi,t\cdot\mathfrak{c}) = t\cdot f(\psi) = t^2\cdot f'(h(\psi)) = f'(h(\psi)) = \text{MNX}(\Gamma'; h(\psi,\mathfrak{c}))$ since $f = t\cdot f'\circ h$ and $t^2 = 1$.

Induction step. Note again that any child of $h(\mathfrak{p})$ must be of the form $h(\mathfrak{p}')$ because h is surjective. If $h(\mathfrak{p}') \leftarrow h(\mathfrak{p})$ then $\mathfrak{p}' \leftarrow \mathfrak{p}$ and therefore $|\mathcal{U}(\mathfrak{p}')| = |\mathcal{U}(\mathfrak{p})| - 1$.

$$\begin{aligned}\text{NGX}(\Gamma'; h(\mathfrak{p})) &= -\min_{h(\mathfrak{p}')\leftarrow h(\mathfrak{p})}\big[\text{NGX}(\Gamma'; h(\mathfrak{p}'))\big] &&\text{(Definition 5.1.3)}\\ &= -\min_{h(\mathfrak{p}')\leftarrow h(\mathfrak{p})}\big[\text{NGX}(\Gamma; t\cdot\mathfrak{p}')\big] &&\text{(induction)}\\ &= -\min_{\mathfrak{p}'\leftarrow\mathfrak{p}}\big[\text{NGX}(\Gamma; t\cdot\mathfrak{p}')\big] &&\text{(h is generation preserving)}\\ &= \text{NGX}(\Gamma; t\cdot\mathfrak{p}) &&\text{(Definition 5.1.3)} \quad \square\end{aligned}$$

Theorem 5.6.2. Let $h : X^{\mathcal{S}} \to X^{\mathcal{S}}$ be a generation preserving anti-automorphism of $\Gamma = \langle X^{\mathcal{S}}, f\rangle$, and let $\mathfrak{p} \in \mathfrak{P}_{\mathcal{S}}$ such that $h(\mathfrak{p}) = \mathfrak{p}$. Then $\text{MNX}(\Gamma;\mathfrak{p}) = -\text{MNX}(\Gamma;-\mathfrak{p})$ and $\text{NGX}(\Gamma;\mathfrak{p}) = \text{NGX}(\Gamma;-\mathfrak{p})$.

Corollary i: If furthermore f is isotone then $\text{NGX}(\Gamma;\mathfrak{p}) = +1$.

Proof. The main theorem is an application of Theorem 5.6.1. The corollary follows from Theorem 5.5.3:iii, where for any $\psi \in X^{\mathcal{S}}$ we have $\text{MNX}(\Gamma; \psi, \text{MAX}) \geqslant \text{MNX}(\Gamma; \psi, \text{MIN}) = -\text{MNX}(\Gamma; \psi, \text{MAX})$ so that $\text{MNX}(\Gamma; \psi, \text{MAX}) \geqslant 0$.[2] Therefore $\text{MNX}(\Gamma; \psi, \text{MAX}) = +1$ and $\text{MNX}(\Gamma; \psi, \text{MIN}) = -1$. □

[2]Note that this can be applied in a more general sense to multi-valued isotone set colouring games to show that, informally speaking, a symmetrical position is at least a draw for the player to move next.

Chapter 6

Metagames

Theorems that prove the minimax value of certain positions often involve one player imagining that the position is somehow altered, then winning the game from the imaginary position, and finding that the actual game has been won also. This section captures such methods in a general theory of metagames. The aim is to determine transformations of colourings that preserve the minimax value while simplifying the practical analysis of the positions.

6.1 Subgames and Supergames

When during the course of a game $\langle X^{\mathcal{S}}, f\rangle$ a position $\mathfrak{p} \in \mathfrak{P}_{\mathcal{S}}$ arises, the players are essentially playing a game on $\mathcal{U}(\mathfrak{p})$ from then on.

Definition 6.1.1. Let $\Gamma = \langle X^{\mathcal{S}}, f\rangle$ and let ψ be some colouring. The **ψ-subfunction** of f is the function $f/\psi : \check{X}^{\mathcal{S}\cap\mathcal{U}(\psi)} \to \mathbb{B}$ defined by

$$\xi \mapsto f\big(\xi\overline{\psi}\big).$$

The game $\Gamma/\psi \overset{\text{def}}{=\!=} \langle X^{\mathcal{S}\setminus\mathcal{A}(\psi)}, f/\psi\rangle$ is the **ψ-subgame** of Γ. For $\mathfrak{m} \in \mathfrak{M}(\mathcal{S})$ and $\mathfrak{c} \in \mathfrak{C}$, write $(\Gamma, \mathfrak{c}) \oplus \mathfrak{m} \overset{\text{def}}{=\!=} (\Gamma/\mathfrak{m}, \bar{\mathfrak{c}})$.

Observation i: If $\mathcal{S} \subseteq \mathcal{A}(\psi)$ then $\Gamma/\psi = \langle\varnothing, f(\psi)\rangle$.

Observation ii: If $\mathcal{A}(\psi) \cap \mathcal{S} = \varnothing$ then $\Gamma/\psi = \Gamma$.

Observation iii: Since the subfunction ignores the uncoloured elements of ψ we have $f/\psi = f/\overline{\psi}$ and therefore also $\Gamma/\psi = \Gamma/\overline{\psi}$.

Example 6.1.2. Let $f : \mathbb{B}^5 \to \mathbb{B}$ and $\psi \in \mathbb{B}^7 = (+, \phi, \phi, +, \phi, -, -)$. Then $f/\psi : \mathbb{B}^{\{1,2,4\}} \to \mathbb{B}$ is the function that maps $\xi \in \mathbb{B}^{\{1,2,4\}}$ to $f\big(+, \xi_1, \xi_2, +, \xi_4\big)$.

So the subgame "fills in" the elements that are coloured in ψ and continues the game from there. When this transformation is performed after every move, then each move essentially leads from an initial position in some game to an initial position in a subgame whose dimension is one less. For this reason we can write $(\Gamma, \mathfrak{c}) \oplus \mathfrak{m} = (\Gamma/\mathfrak{m}, \bar{\mathfrak{c}})$. The equivalence of each position in a set colouring game to the empty position in another set colouring game is confirmed in the following theorem.

Theorem 6.1.3. Let $\Gamma = \langle X^{\mathcal{S}}, f \rangle$ with $\psi \in X^{\mathcal{S}}$, and let $\Gamma' = \langle X^{\mathcal{S}'}, f' \rangle = \Gamma/\psi$ with $\mathfrak{p}' \in \mathfrak{P}_{\mathcal{S}'}$. Then $\text{MNX}(\Gamma'; \mathfrak{p}') = \text{MNX}(\Gamma; \mathfrak{p}'\overline{\psi})$.

Corollary i: Let $\mathfrak{p} = (\psi, \mathfrak{c}) \in \mathfrak{P}_{\mathcal{S}}$ then $\text{MNX}(\Gamma; \mathfrak{p}) = \text{MNX}(\Gamma/\psi; \mathfrak{c})$.

Due to this theorem, the terms *position* and *game* are essentially interchangeable. The discussion in the following chapters will therefore concentrate only on empty positions. The methods can be applied to any position by considering the corresponding subgame.

Given a particular game $\langle X^{\mathcal{S}}, f \rangle$, and a particular move $\mathfrak{m}$ outside of $\mathcal{S}$, it is possible to construct a larger game Γ' such that $\Gamma = \Gamma'/\mathfrak{m}$:

Theorem 6.1.4. Let $\Gamma = \langle X^{\mathcal{S}}, f \rangle$, let $\mathfrak{m} = \chi^v$ with $v \notin \mathcal{S}$, and let $X' \subseteq X$ such that $\chi \in X'$. Consider the game

$$\Gamma' = \left\langle X^{\mathcal{S}+v}, \xi \mapsto f(\xi) \wedge (\xi_v \in X') \right\rangle.$$

For this game we have $\Gamma'/\mathfrak{m} = \Gamma$.

It should be noted that not every game Γ' satisfying $\Gamma'/\mathfrak{m} = \Gamma$ needs to be of this form, as the outcome functions of Γ' only needs to agree with f for pure colourings $\psi^* \in \check{X}^{\mathcal{S}+v}$ with $\psi^*_v = \chi$. Modifying the outcome function for Γ' in any arbitrary way whenever $\psi^*_v \neq \chi$ still yields a game with the desired property.

The previous construction is an example of the opposite of a subgame. Where a subgame is played on a subset of the original game's colour space, the game generated in Theorem 6.1.4 is played on a superset thereof. Another construction of this form will play an important role in the remainder of this thesis:

Definition 6.1.5. Let $\Gamma = \langle X^{\mathcal{S}}, f \rangle$, and $\mathcal{S}^* \supseteq \mathcal{S}$. The game $\langle X^{\mathcal{S}^*}, f \rangle$ is the **supergame** of Γ on the set $\mathcal{S}^*$. This supergame is denoted $\Gamma * \mathcal{S}^*$.

The reason for this notation is that $\Gamma * \mathcal{S}^*$ is created from Γ by adding a set $\mathcal{S}^* \setminus \mathcal{S}$ of dead elements. Therefore from Theorem 5.4.2 it is evident that the outcome with optimal play of $\Gamma * \mathcal{S}^*$ is equal to the outcome of Γ if the added number of elements is even, and equal to the outcome of $\Gamma*$ otherwise.

Note that it is crucial for this definition that $\mathcal{S}^* \supseteq \mathcal{S}$, for if there were an element $v \in \mathcal{S} \setminus \mathcal{S}^*$ then this element would be uncoloured when a pure colouring of $\mathcal{S}^*$ is projected onto $\mathcal{S}$, and f would not be able to assign a value since it is only defined for pure colourings of $\mathcal{S}$. This is the reason that subgames must be defined using a colouring of the "dropped" elements.

According to Theorem 5.4.2 the only thing that really matters about $\mathcal{S}^*$ in $\Gamma * \mathcal{S}^*$ is its parity.

Therefore the following shorthand notation is used.

Definition 6.1.6. Let $\Gamma = \langle X^{\mathcal{S}}, f \rangle$. Then $\Gamma * \Box$ refers to $\Gamma * \mathcal{S}^*$ for some arbitrary $\mathcal{S}^* \supseteq \mathcal{S}$ where $|\mathcal{S}^*|$ is even, and $\Gamma * \triangle$ refers to $\Gamma * \mathcal{S}^*$ for some arbitrary $\mathcal{S}^* \supseteq \mathcal{S}$ where $|\mathcal{S}^*|$ is odd.

So if $|\mathcal{S}|$ itself is even then $\Gamma * \Box = \Gamma = (\Gamma * \triangle)*$, and if $|\mathcal{S}|$ is odd then $\Gamma * \triangle = \Gamma = (\Gamma * \Box)*$.

6.2 Metagames

Game transformations that reduce the complexity of the analysis often involve breaking down the game into several independent local games. This is similar to what happens in Combinatorial Game Theory (CGT) [23, 13]; however, most CGT results relate to the goal of being the *last* player to move. This goal is irrelevant in set colouring games, as the number of moves available to each player trivially decreases by exactly one with every move.

Definition 6.2.1. Let $\mathcal{Q} = \{\langle X^{\mathcal{S}_0}, f_0 \rangle, \langle X^{\mathcal{S}_1}, f_1 \rangle, \ldots, \langle X^{\mathcal{S}_{k-1}}, f_{k-1} \rangle\}$ be a family of games, and $f : \mathbb{B}^k \to \mathbb{B}$. Put $\mathcal{S} \overset{\text{def}}{=\!=} \bigcup_{i \in \mathbb{Z}_k} \mathcal{S}_i$. The **metagame** $\langle\langle \mathcal{Q}, f \rangle\rangle$ is the game

$$\langle X^{\mathcal{S}}, f \circ (f_i)_{i \in \mathbb{Z}_k} \rangle. \tag{6.2.1}$$

The function f is its **metafunction**, and the games $\langle X^{\mathcal{S}_i}, f_i \rangle$ are its **component games**. The function $(f_i)_{i \in \mathbb{Z}_k} : \mathbb{B}^{\mathcal{S}} \to \mathbb{B}^k$ returns the **result vector** of a given complete colouring under $\langle\langle \mathcal{Q}, f \rangle\rangle$.

The result vector contains the results of all the component games for any complete colouring in $\mathbb{B}^{\mathcal{S}}$. The sets $\mathcal{S}_i$ do not need to be pairwise disjoint.

Example 6.2.2. The game of Y can be seen as a metagame. Consider $f_{\mathbb{Y}_2} : \mathbb{B}^{\mathbb{Y}_2} \to \mathbb{B}$. This function is equivalent to the metafunction defined by the following equations:

$$\begin{aligned}
f_0 &: \xi \mapsto (\xi_{0,0,2} \wedge \xi_{1,0,1}) \vee (\xi_{0,0,2} \wedge \xi_{0,1,1}) \vee (\xi_{1,0,1} \wedge \xi_{0,1,1}), \\
f_1 &: \xi \mapsto (\xi_{1,0,1} \wedge \xi_{2,0,0}) \vee (\xi_{1,0,1} \wedge \xi_{1,1,0}) \vee (\xi_{2,0,0} \wedge \xi_{1,1,0}), \\
f_2 &: \xi \mapsto (\xi_{0,1,1} \wedge \xi_{1,1,0}) \vee (\xi_{0,1,1} \wedge \xi_{0,2,0}) \vee (\xi_{1,1,0} \wedge \xi_{0,2,0}), \\
f &: \xi \mapsto (\xi_0 \wedge \xi_1) \vee (\xi_0 \wedge \xi_2) \vee (\xi_1 \wedge \xi_2).
\end{aligned}$$

Given the pure colouring ψ with $\psi_{0,1,1} = \psi_{0,2,0} = \psi_{1,0,1} = \text{T}$ and $\psi_{0,0,2} = \psi_{1,1,0} = \psi_{2,0,0} = \text{F}$, we obtain the result vector $(f_0(\psi), f_1(\psi), f_2(\psi)) = (+, -, +)$ and the outcome of the metagame is $f(+, -, +) = +1$.

Definition 6.2.3. Let $\mathcal{Q} = \{\Gamma_i\}_{i \in \mathbb{Z}_k}$ and $\Gamma_i = \langle X^{\mathcal{S}_i}, f_i \rangle$ with $\mathcal{S} = \bigcup_{i \in \mathbb{Z}_k} \mathcal{S}_i$. The following terminology is used:

- Given a metagame $\langle\langle \mathcal{Q}, f \rangle\rangle$, the game $\Gamma_i * \mathcal{S}$ is the **embedded component** of Γ_i in $\langle\langle \mathcal{Q}, f \rangle\rangle$.
- If $\mathcal{S}_i \cap \mathcal{S}_j = \varnothing$, then Γ_i and Γ_j are **independent** components.

- If all components are pairwise independent, so that $\{\mathcal{S}_i\}_{i\in\mathbb{Z}_k}$ is a partition, then $\langle\langle\mathcal{Q}, f\rangle\rangle$ is **a partition game**.
- If $f : \xi \mapsto \bigwedge_{i\in\mathbb{Z}_k} \xi_i$ then $\langle\langle\mathcal{Q}, f\rangle\rangle$ is **a conjunctive metagame**, denoted as $\bigwedge_i \Gamma_i$ or $\langle\langle\mathcal{Q}, \wedge\rangle\rangle$.
- If $f : \xi \mapsto \bigvee_{i\in\mathbb{Z}_k} \xi_i$ then $\langle\langle\mathcal{Q}, f\rangle\rangle$ is **a disjunctive metagame**, denoted as $\bigvee_i \Gamma_i$ or $\langle\langle\mathcal{Q}, \vee\rangle\rangle$.

Thus a partition game is a game that can be decomposed into several *independent* component games, with conjunctive and disjunctive partition games being the most common cases.

Theorem 6.2.4. Every disjunctive metagame $\bigvee_i \langle X^{\mathcal{S}_i}, f_i\rangle$ has a dual representation as a conjunctive metagame.

In fact each disjunctive metagame can be turned into a conjunctive metagame, and vice versa, by negating the metafunction as well as all the component functions; see the proof on page 61.

Theorem 6.2.5. Every metagame where one of the players attempts to make the result vector equal to ψ for some $\psi \in \mathbb{B}^{\mathbb{Z}_k}$ has a representation as a conjunctive metagame.

This also implies that any metagame in which one of the players tries to *avoid* the result vector equalling ψ is a conjunctive metagame. Thus any theorems about conjunctive metagames apply to all games of this type.

A supergame can be regarded as a specific kind of metagame:

Theorem 6.2.6. Let $\Gamma = \langle X^{\mathcal{S}}, f\rangle$ and $\mathcal{S}^* \supseteq \mathcal{S}$. Then $\Gamma * \mathcal{S}^* = \Gamma \wedge \langle X^{\mathcal{S}^*\setminus\mathcal{S}}, +\rangle = \Gamma \vee \langle X^{\mathcal{S}^*\setminus\mathcal{S}}, -\rangle$.

Another straightforward assertion is that a subgame of a metagame is equal to the metagame of the subgames of the components.

Theorem 6.2.7. Let $\Gamma = \langle\langle\{\Gamma_i\}_{i\in\mathbb{Z}_k}, f\rangle\rangle$ with $\Gamma_i = \langle X^{\mathcal{S}_i}, f_i\rangle$ and $\mathcal{S} = \bigcup_{i\in\mathbb{Z}_k} \mathcal{S}_i$. Let $\psi \in X^{\mathcal{S}}$. Then $\Gamma/\psi = \langle\langle\{\Gamma_i/\psi\}_{i\in\mathbb{Z}_k}, f\rangle\rangle$.

6.3 Comparing Games

When two set colouring games $\langle X^{\mathcal{S}}, f\rangle$ and $\langle X^{\mathcal{S}}, f'\rangle$ are defined on the same colour space and it turns out that $f(\psi^*) \geqslant f'(\psi^*)$ for every complete colouring ψ^* of $\mathcal{S}$, then of course $\langle X^{\mathcal{S}}, f\rangle$ is more advantageous to MAX and $\langle X^{\mathcal{S}}, f'\rangle$ is more advantageous to MIN. This allows a partial ordering of games defined on the same colour space.

Definition 6.3.1. Let $\Gamma = \langle X^{\mathcal{S}}, f\rangle$ and $\Gamma' = \langle X^{\mathcal{S}}, f'\rangle$. If $f'(\psi^*) \geqslant f(\psi^*)$ for all complete colourings $\psi^* \in \check{X}^{\mathcal{S}}$, then Γ' is **Γ-necessary** and Γ is **Γ'-sufficient**. This is denoted as $\Gamma' \geqslant \Gamma$. Put $\Gamma' = \Gamma$ if and only if $\Gamma' \geqslant \Gamma$ and $\Gamma' \leqslant \Gamma$.

Observation i: By induction it follows that if $\Gamma' \geqslant \Gamma$ then $\text{MNX}(\Gamma'; \mathfrak{p}) \geqslant \text{MNX}(\Gamma; \mathfrak{p})$ for all $\mathfrak{p} \in \mathfrak{P}_{\mathcal{S}}$.

Observation ii: Let $\psi \in X^{\mathcal{S}}$. If $\Gamma' \geqslant \Gamma$ then it can be easily verified from Definition 5.1.4 that $\mathfrak{M}^+_{\Gamma}(\psi, \text{MAX}) \subseteq \mathfrak{M}^+_{\Gamma'}(\psi, \text{MAX})$ and $\mathfrak{M}^+_{\Gamma}(\psi, \text{MIN}) \supseteq \mathfrak{M}^+_{\Gamma'}(\psi, \text{MIN})$.

This is a partial order; it is possible for games on the same colour space to be incomparable. As the notation suggests, when $\Gamma' \leqslant \Gamma$ then for Γ to be positive it is sufficient if Γ' is positive. If that is indeed the case, then MAX can win Γ by using a winning strategy from Γ'. Conversely, for Γ' to be negative it suffices if Γ is negative, and MIN can win Γ' by using a winning strategy from Γ. If $\Gamma' \leqslant \Gamma$ and $\Gamma' \geqslant \Gamma$ then $f = f'$, so it then indeed makes sense to write $\Gamma' = \Gamma$.

The partial order can be extended to compare games that are not defined on the same colour space, by comparing supergames that *are* defined on the same colour space.

Definition 6.3.2. Let $\Gamma = \langle X^{\mathcal{S}}, f \rangle$ and $\Gamma' = \langle X^{\mathcal{S}'}, f' \rangle$. Then $\Gamma' \geqslant \Gamma$ if and only if $\Gamma' * \mathcal{S}^* \geqslant \Gamma * \mathcal{S}^*$ where $\mathcal{S}^* = \mathcal{S} \cup \mathcal{S}'$.

Observation i: Equivalently, $\Gamma' \geqslant \Gamma$ if and only if $f'(\psi) \geqslant f(\psi)$ for all $\psi \in X^{\mathcal{S} \cup \mathcal{S}'}$.

Observation ii: If $\Gamma = \Gamma'$ then $\mathcal{S} \setminus \mathcal{S}'$ is dead in Γ and $\mathcal{S}' \setminus \mathcal{S}$ is dead in Γ'.

Observation iii: For any two games Γ and Γ' we have $\Gamma \vee \Gamma' \geqslant \Gamma \geqslant \Gamma \wedge \Gamma'$.

Observation iv: For any $\Diamond \in \{\Box, \triangle\}$ we have $\Gamma \geqslant \Gamma' \iff \Gamma * \Diamond \geqslant \Gamma' * \Diamond$.

Observation v: $\langle \varnothing, + \rangle \geqslant \Gamma \geqslant \langle \varnothing, - \rangle$.

Comparing two games entails "padding" both with dead elements until the colour spaces match. While the added components are trivial by themselves, they are not entirely pointless in a metagame:

Theorem 6.3.3. Let $\Gamma = \langle X^{\mathcal{S}}, f \rangle$ and $\mathcal{S}^* \subseteq \mathcal{S}$, then

$$\Gamma * \mathcal{S}^* = \begin{cases} \Gamma & \text{if } |\mathcal{S}^* \setminus \mathcal{S}| \text{ is even,} \\ \Gamma * & \text{if } |\mathcal{S}^* \setminus \mathcal{S}| \text{ is odd.} \end{cases}$$

Corollary i: $(\Gamma_0 * \Box) \wedge (\Gamma_1 * \Box) = (\Gamma_0 * \triangle) \wedge (\Gamma_1 * \triangle) = \Gamma_0 \wedge \Gamma_1 * \Box$, and the same goes for $\vee$ instead of $\wedge$;

Corollary ii: $(\Gamma_0 * \Box) \wedge (\Gamma_1 * \triangle) = (\Gamma_0 * \triangle) \wedge (\Gamma_1 * \Box) = \Gamma_0 \wedge \Gamma_1 * \triangle$, and the same goes for $\vee$ instead of $\wedge$.

Corollary iii: Let $\Diamond \in \{\Box, \triangle\}$ and $\mathfrak{m} \in \mathfrak{M}(\mathcal{S})$, then $(\Gamma * \Diamond) \oplus \mathfrak{m} = (\Gamma \oplus \mathfrak{m}) * \Diamond *$.

All this immediately follows from Theorem 5.4.2, by considering the number of stars present in each game. From there we have:

Theorem 6.3.4. Let $\Gamma = \langle X^{\mathcal{S}}, f\rangle$ and $\Gamma' = \langle X^{\mathcal{S}'}, f'\rangle$ with $\Gamma \leqslant \Gamma'$, and let $\mathfrak{c} \in \mathfrak{C}$. If $|\mathcal{S}\backslash\mathcal{S}'|$ and $|\mathcal{S}'\backslash\mathcal{S}|$ are even then $\text{MNX}(\Gamma; \mathfrak{c}) = +1 \implies \text{MNX}(\Gamma'; \mathfrak{c}) = +1$ and $\text{MNX}(\Gamma'; \mathfrak{c}) = -1 \implies \text{MNX}(\Gamma; \mathfrak{c}) = -1$. If Γ and Γ' are isotone then the requirement that $|\mathcal{S} \setminus \mathcal{S}'|$ and $|\mathcal{S}' \setminus \mathcal{S}|$ be even can be dropped.

Game comparison has the desired property that adding a conjunctive or disjunctive component preserves the comparison:

Theorem 6.3.5. Let $\Gamma_0 = \langle X^{\mathcal{S}_0}, f_0\rangle$ and $\Gamma_1 = \langle X^{\mathcal{S}_1}, f_1\rangle$ such that $\Gamma_0 \leqslant \Gamma_1$. For any $\Gamma_2 = \langle X^{\mathcal{S}}, f\rangle$ we then have $\Gamma_0 \wedge \Gamma_2 \leqslant \Gamma_1 \wedge \Gamma_2$ and $\Gamma_0 \vee \Gamma_2 \leqslant \Gamma_1 \vee \Gamma_2$.

Another useful observation is that if two games are both necessary or both sufficient for a third game, then so are their conjunction and disjunction:

Theorem 6.3.6. Let $\Gamma_i = \langle X^{\mathcal{S}_i}, f_i\rangle$ for $0 \leqslant i \leqslant 2$. If $\Gamma_0 \geqslant \Gamma_2$ and $\Gamma_1 \geqslant \Gamma_2$, then $\Gamma_0 \vee \Gamma_1 \geqslant \Gamma_0 \wedge \Gamma_1 \geqslant \Gamma_2$. If $\Gamma_0 \leqslant \Gamma_2$ and $\Gamma_1 \leqslant \Gamma_2$, then $\Gamma_0 \wedge \Gamma_1 \leqslant \Gamma_0 \vee \Gamma_1 \leqslant \Gamma_2$.

Adding a star to any component of a metagame is the same as adding a star to the entire metagame:

Theorem 6.3.7. Let $\mathcal{Q} = \{\Gamma_i\}_{i\in\mathbb{Z}_k}$. Put $\mathcal{Q}' = \{\Gamma_0 *, \Gamma_1, \Gamma_2, \ldots, \Gamma_{k-1}\}$. Then $\langle\langle \mathcal{Q}', f\rangle\rangle = \langle\langle \mathcal{Q}, f\rangle\rangle *$.

Comparing two games can be useful when one game is easier to analyze computationally than the other. The next two chapters will deal with mappings that can transform a set colouring game into another set colouring game that has lower dimension and is provably sufficient or necessary.

6.4 Values for Conjunctive and Disjunctive Metagames

As mentioned in Section 6.1, without loss of generality only initial positions need to be considered when analyzing metagames. The question at hand is: Can the minimax value of a metagame be determined by analyzing the component games in isolation? In many cases this is indeed possible.

Theorem 6.4.1. Let $\mathcal{Q} = \{\langle X^{\mathcal{S}_i}, f_i\rangle\}_{i\in\mathbb{Z}_k}$ be a partition game with $\mathcal{S} = \bigcup_{i\in\mathbb{Z}_k} \mathcal{S}_i$. Then

$$\langle\langle \mathcal{Q}, \vee\rangle\rangle \;\leqslant\; \Gamma_i \;\leqslant\; \langle\langle \mathcal{Q}, \wedge\rangle\rangle$$

for all $i \in \mathbb{Z}_k$.

Corollary i: For any $\mathfrak{c} \in \mathfrak{C}$ and $\Diamond \in \{\Box, \triangle\}$, if $\exists_{i\in\mathbb{Z}_k}\big[\text{MNX}(\Gamma_i * \Diamond; \mathfrak{c}) = +1\big]$ then $\text{MNX}\big(\langle\langle \mathcal{Q}, \vee\rangle\rangle * \Diamond; \mathfrak{c}\big) = +1$.

Corollary ii: For any $\mathfrak{c} \in \mathfrak{C}$ and $\Diamond \in \{\Box, \triangle\}$, if $\exists_{i\in\mathbb{Z}_k}\big[\text{MNX}(\Gamma_i * \Diamond; \mathfrak{c}) = -1\big]$ then $\text{MNX}\big(\langle\langle \mathcal{Q}, \wedge\rangle\rangle * \Diamond; \mathfrak{c}\big) = -1$.

In these cases there exists a winning strategy that concentrates on winning just one component. The theorem therefore even holds when the component games are not independent. It is necessary for MAX to be able to win each of the embedded components in order to win the conjunctive metagame.

This is not in general sufficient for MAX; for instance, when $\Gamma_0 = \langle \mathbb{T}^{\{0,1,2\}}, \xi \mapsto \xi_0 \vee \xi_1 \vee \xi_2 \rangle$ and $\Gamma_1 = \langle \mathbb{T}^{\{3,4\}}, \xi \mapsto \xi_3 \equiv \xi_4 \rangle$, with MIN having the first move, MAX wins both $\Gamma_0 * \triangle$ and $\Gamma_0 * \triangle$ but not $\Gamma_0 \wedge \Gamma_1$. The question thus arises what circumstances are required for sufficiency.

Theorem 6.4.2. Let $\mathcal{Q} = \{\langle X^{\mathcal{S}_i}, f_i \rangle\}_{i \in \mathbb{Z}_k}$ be a partition game with $\mathcal{S} = \bigcup_{i \in \mathbb{Z}_k} \mathcal{S}_i$. Then:

$$\forall_{i \in \mathbb{Z}_k} \left[\text{MNX}(\Gamma_i * \Box; \text{MIN}) = +1 \right] \Longrightarrow \text{MNX}\big(\langle\langle \mathcal{Q}, \wedge \rangle\rangle * \Box; \text{MIN} \big) = +1;$$

$$\forall_{i \in \mathbb{Z}_k} \left[\text{MNX}(\Gamma_i * \Box; \text{MAX}) = -1 \right] \Longrightarrow \text{MNX}\big(\langle\langle \mathcal{Q}, \vee \rangle\rangle * \Box; \text{MAX} \big) = -1;$$

Combining Theorems 6.4.1 and 6.4.2 we have that, in an even or isotone metagame, a second-player win in each embedded component is both necessary and sufficient for a second-player win in all the components. If all components are even, then a simple strategy can be used where each move by the opponent is answered with a move in the same component. The proof of Theorem 6.4.2 as outlined on page 63 uses this approach. Such a strategy may be called **a partition strategy**; an example already surfaced earlier, in the proof of Theorem 5.4.2 which may indeed be regarded as a special case of Theorem 6.4.2 since $\Gamma ** = \Gamma \wedge \langle X^2, + \rangle$.

Strictly speaking a partition strategy is not a strategy according to Definition 3.3.1, since its domain is not $X^{\mathcal{S}}$ but rather $\mathfrak{M}(\phi^{\mathcal{S}}) \times X^{\mathcal{S}}$, a subset of $X^{\mathcal{S}} \times X^{\mathcal{S}}$. Moreover, a partition strategy is powerless to decide in situations where there is no previous move by the opponent, or when the opponent took the last available move in some component.

For this reason, there is no component-wise partition strategy when there are odd components. Whereas a winning strategy for Γ can easily be translated into a winning strategy for $\Gamma **$, the reverse is not necessarily the case. Nevertheless, when the conditions of Theorem 6.4.2 are met there is a winning strategy that is almost as good as a pure partition strategy. The winner can choose to pair up all the odd components, treating each pair as an even component. Therefore there will be a winning strategy in which each move in an even component is answered with a move in the same component, and each move in an odd component is answered with a move in either the same component or the one that was paired up with it.

Theorem 6.4.1 can be "backed up" one move to give the following:

Theorem 6.4.3. Let $\mathcal{Q} = \{\langle X^{\mathcal{S}_i}, f_i \rangle\}_{i \in \mathbb{Z}_k}$ be a family of independent games, and let $\Diamond \in \{\Box, \triangle\}$. If there exists a Γ_j such that $\text{MNX}(\Gamma_j * \Diamond; \text{MIN}) = -1$, then any $\mathfrak{m} \notin \mathfrak{M}(\mathcal{S}_j)$ is a losing move in $(\langle\langle \mathcal{Q}, \wedge \rangle\rangle * \Diamond *, \text{MAX})$. If there exists a Γ_j such that $\text{MNX}(\Gamma_j * \Diamond; \text{MAX}) = +1$, then any $\mathfrak{m} \notin \mathfrak{M}(\mathcal{S}_j)$ is a losing move in $(\langle\langle \mathcal{Q}, \vee \rangle\rangle * \Diamond *, \text{MIN})$.

> *Corollary i:* Let Γ_i and Γ_j be two components such that $i \neq j$. If $\text{MNX}(\Gamma_i * \Diamond; \text{MIN}) = -1$ and $\text{MNX}(\Gamma_j * \Diamond; \text{MIN}) = -1$, then $\text{MNX}(\langle\langle \mathcal{Q}, \wedge \rangle\rangle * \Diamond *; \text{MAX}) = -1$. If $\text{MNX}(\Gamma_i * \Diamond; \text{MAX}) = +1$ and $\text{MNX}(\Gamma_j * \Diamond; \text{MAX}) = +1$, then $\text{MNX}(\langle\langle \mathcal{Q}, \vee \rangle\rangle * \Diamond *; \text{MAX}) = +1$.

The justification for the main theorem is that $\mathfrak{m} \notin \mathfrak{M}(\mathcal{S}_j)$ sets up the preconditions of Theorem 6.4.1, so that the player to move is forced to move in $\mathcal{S}_j$. The corollary then follows because if there are two such components then the player to move cannot move in both of them simultaneously, as all components are independent.

Theorem 6.4.2 can similarly be backed up one move:

Theorem 6.4.4. Let $\mathcal{Q} = \{\langle X^{\mathcal{S}_i}, f_i\rangle\}_{i\in\mathbb{Z}_k}$ be set of independent games. If there are no empty components, then:

$$\exists_{i\in\mathbb{Z}_k}\big[\text{MNX}(\Gamma_i * \triangle; \text{MAX}) = +1\big] \wedge \forall_{j\in\mathbb{Z}_k\setminus i}\big[\text{MNX}(\Gamma_i * \square; \text{MIN}) = +1\big] \implies \text{MNX}\big(\langle\langle\mathcal{Q}, \wedge\rangle\rangle * \triangle; \text{MAX}\big) = +1;$$

$$\exists_{i\in\mathbb{Z}_k}\big[\text{MNX}(\Gamma_i * \triangle; \text{MIN}) = -1\big] \wedge \forall_{j\in\mathbb{Z}_k\setminus i}\big[\text{MNX}(\Gamma_i * \square; \text{MAX}) = -1\big] \implies \text{MNX}\big(\langle\langle\mathcal{Q}, \vee\rangle\rangle * \triangle; \text{MIN}\big) = -1.$$

The winning opening move in both cases is to play a winning move in $\Gamma_i * \triangle$, creating a situation in which Theorem 6.4.2 applies. The theorem also holds when Γ_i is even, with counterintuitive consequences. The game $\langle\langle\mathcal{Q}, \wedge\rangle\rangle * *$ can be won by MAX going first, by playing a winning opening move in $\Gamma_i *$. However, this move might *not* be in Γ_i itself, for it might require MAX to play in the star of $\Gamma_i *$. If that is the case, then this opening move for $\langle\langle\mathcal{Q}, \wedge\rangle\rangle * *$ does not work for $\langle\langle\mathcal{Q}, \wedge\rangle\rangle$ itself. Yet there must be a winning move in $\langle\langle\mathcal{Q}, \wedge\rangle\rangle$, otherwise there would be no winning move in $\langle\langle\mathcal{Q}, \wedge\rangle\rangle * *$. This winning move $\mathfrak{m}$ in $\langle\langle\mathcal{Q}, \wedge\rangle\rangle$ must then be in some other component Γ_i. After this move, the game is won for MAX even though it appears to contain two components whose status is unknown, since nothing was specified about $\text{MNX}(\Gamma_i * \mathcal{S}*; \text{MIN})$ or $\text{MNX}(\Gamma_i/\mathfrak{m} * \mathcal{S}; \text{MIN})$.

The discussion in this section has concentrated on partition games, with results for second-player wins in even games and first-player wins in odd games. This leaves the following situations to be considered:

- the components are not independent;
- the metagame has the "wrong parity".

If the components are not independent then there can be no general analysis that always settles the minimax value of the metagame based only on the analyses of the individual components, since examples can easily be constructed in which the outcome depends crucially on the overlap between components. For instance, if $\Gamma_0 = \langle\mathbb{T}^{\mathcal{S}_0}, \xi \mapsto \xi_v\rangle$ and $\Gamma_1 = \langle\mathbb{T}^{\mathcal{S}_1}, \xi \mapsto \xi_w\rangle$, then $\Gamma_0 \wedge \Gamma_1$ is negative if $v \neq w$ but regular if $v = w$.

The remaining case is the "off-parity metagame", namely when the player who needs to win all the components has the first move in an even game or the second move in an odd game; in other words, if this player does not have the *last* move of the game. Chapter 7 will delve deeper into the question of determining the minimax value of a metagame by independent analysis of the components. However, as in previous cases, isotone games are much tamer.

6.5 Isotone Metagames

If all components are isotone then the behaviour is more benign, since there are no misère colourings in an isotone function. Another way of saying this is that stars can be added at will to isotone components, making each of them even as desired.

Theorem 6.5.1. Let $\mathcal{Q} = \{\Gamma_i\}_{i \in \mathbb{Z}_k}$ be a family of isotone pairwise independent games. Let $n_{\rm P}$, $n_{\rm R}$, and $n_{\rm N}$ be the number of positive, regular, and negative components.

$$\langle\langle \mathcal{Q}, \wedge \rangle\rangle \text{ is } \begin{cases} \text{positive} & \text{if } n_{\rm R} = 0 \text{ and } n_{\rm N} = 0; \\ \text{regular} & \text{if } n_{\rm R} = 1 \text{ and } n_{\rm N} = 0; \\ \text{negative} & \text{if } n_{\rm R} \geqslant 2 \text{ or } n_{\rm N} \geqslant 1. \end{cases}$$

$$\langle\langle \mathcal{Q}, \vee \rangle\rangle \text{ is } \begin{cases} \text{negative} & \text{if } n_{\rm R} = 0 \text{ and } n_{\rm P} = 0; \\ \text{regular} & \text{if } n_{\rm R} = 1 \text{ and } n_{\rm P} = 0; \\ \text{positive} & \text{if } n_{\rm R} \geqslant 2 \text{ or } n_{\rm P} \geqslant 1. \end{cases}$$

These values can be appreciated intuitively. If all components are positive then MAX can win each component separately, so MAX could win the metagame by using a partition strategy if MAX were allowed to skip. But an isotone game that can be won with skips can also be won without skips.

If there is one regular component and the rest are positive, then MAX wins by playing a winning move in the regular component. This component then becomes positive, since misère isotone games do not exist. If however MIN has the first move, then MIN can win the regular component and so win the conjunctive metagame.

6.6 Proofs

Theorem 6.1.3. Let $\Gamma = \langle X^{\mathcal{S}}, f \rangle$ with $\psi \in X^{\mathcal{S}}$, and let $\Gamma' = \langle X^{\mathcal{S}'}, f' \rangle = \Gamma/\psi$ with $\mathfrak{p}' \in \mathfrak{P}_{\mathcal{S}'}$. Then $\text{MNX}(\Gamma'; \mathfrak{p}') = \text{MNX}(\Gamma; \mathfrak{p}'\overline{\psi})$.

Corollary i: Let $\mathfrak{p} = (\psi, \mathfrak{c}) \in \mathfrak{P}_{\mathcal{S}}$ then $\text{MNX}(\Gamma; \mathfrak{p}) = \text{MNX}(\Gamma/\psi; \mathfrak{c})$.

Proof. Induction to $|\mathcal{U}(\mathfrak{p}')|$. Let $\mathfrak{p}' = (\psi', \mathfrak{c})$.

Base case: $\mathcal{U}(\mathfrak{p}') = \varnothing$. Then $\text{MNX}(\Gamma; \mathfrak{p}'\overline{\psi}) = \text{MNX}(\Gamma; \psi'\overline{\psi}, \mathfrak{c}) = \text{MNX}(\Gamma'; \mathfrak{p}') = f'(\psi') = f(\psi'\overline{\psi})$ since both $\mathfrak{p}'$ and $\mathfrak{p}'\overline{\psi}$ are final positions.

Induction step. Let $\mathfrak{p}'' \in \mathfrak{P}_{\mathcal{S}'}$. From Definitions 2.2.1 and 3.2.4 it can be verified easily that $\mathfrak{p}''\overline{\psi} \in \mathfrak{P}_{\mathcal{S}}$ and that $\mathfrak{p}'' \leftarrow \mathfrak{p}'$ if and only if $\mathfrak{p}''\overline{\psi} \leftarrow \mathfrak{p}'\overline{\psi}$. Therefore

$$\begin{aligned} \text{NGX}(\Gamma'; \mathfrak{p}') &= -\min_{\mathfrak{p}'' \leftarrow \mathfrak{p}'} [\text{NGX}(\Gamma'; \mathfrak{p}'')] && \text{(Observation 5.1.3:i)} \\ &= -\min_{\mathfrak{p}'' \leftarrow \mathfrak{p}'} [\text{NGX}(\Gamma; \mathfrak{p}''\overline{\psi})] && \text{(induction, } |\mathcal{U}(\mathfrak{p}'')| < |\mathcal{U}(\mathfrak{p}')|) \\ &= -\min_{\mathfrak{p}''\overline{\psi} \leftarrow \mathfrak{p}'\overline{\psi}} [\text{NGX}(\Gamma; \mathfrak{p}''\overline{\psi})] && (\mathfrak{p}'' \leftarrow \mathfrak{p}' \iff \mathfrak{p}''\overline{\psi} \leftarrow \mathfrak{p}'\overline{\psi}) \\ &= \text{NGX}(\Gamma; \mathfrak{p}'\overline{\psi}) && \text{(Observation 5.1.3:i)} \end{aligned}$$

which is equivalent to $\text{MNX}(\Gamma'; \mathfrak{p}') = \text{MNX}(\Gamma; \mathfrak{p}'\overline{\psi})$. □

Theorem 6.1.4. Let $\Gamma = \langle X^{\mathcal{S}}, f\rangle$, let $\mathfrak{m} = \chi^v$ with $v \notin \mathcal{S}$, and let $X' \subseteq X$ such that $\chi \in X'$. Consider the game

$$\Gamma' = \left\langle X^{\mathcal{S}+v}, \xi \mapsto f(\xi) \wedge (\xi_v \in X')\right\rangle.$$

For this game we have $\Gamma'/\mathfrak{m} = \Gamma$.

Proof. Let f' be the outcome function of Γ' as defined in the construction. The game $\Gamma'/\mathfrak{m}$ is played on $X^{(\mathcal{S}+v)-v} = X^{\mathcal{S}}$, with outcome function $f'/\mathfrak{m} \overset{\text{def}}{=\!=} \xi \mapsto f'(\xi\mathfrak{m})$. It remains to show that $f'/\mathfrak{m} = f$. For $\psi^* \in \check{X}^{\mathcal{S}}$ we have $f(\psi^*\mathfrak{m}) \overset{\text{def}}{=\!=} f\big((\psi^*\mathfrak{m}) \searrow \mathcal{S}\big) = f(\psi^*)$ by Lemma 2.2.2:iii. Therefore

$$f'(\psi^*\mathfrak{m}) = f(\psi^*\mathfrak{m}) \wedge \big((\psi^*\mathfrak{m})_v \in X'\big) = f(\psi^*) \wedge \text{TRUE} = f(\psi^*)$$

since $(\psi^*\mathfrak{m})_v = \mathfrak{m}_v = \chi$ because $v \in \mathcal{D}(\mathfrak{m})$. □

Theorem 6.2.4. Every disjunctive metagame $\bigvee\langle X^{\mathcal{S}_i}, f_i\rangle$ has a dual representation as a conjunctive metagame.

Proof. Since $\bigvee\langle X^{\mathcal{S}_i}, f_i\rangle$ uses the metafunction $f : \xi \mapsto \bigvee_{i\in\mathbb{Z}_k} \xi_i$, we have

$$f \circ (f_i)_{i\in\mathbb{Z}_k} = \bigvee_{i\in\mathbb{Z}_k} f_i = -\bigwedge_{i\in\mathbb{Z}_k} -f_i$$

which describes a conjunctive metagame. □

Theorem 6.2.5. Every metagame where one of the players attempts to make the result vector equal to ψ for some $\psi \in \mathbb{B}^{\mathbb{Z}_k}$ has a representation as a conjunctive metagame.

Proof. The metagame in which MAX attempts to make the result vector equal to ψ uses the metafunction $f : \xi \mapsto \bigwedge_{v\in\mathcal{S}} \psi_v \cdot \xi_v$ according to Lemma 2.3.1. Then:

$$f \circ (f_i)_{i\in\mathbb{Z}_k} = \bigwedge_{i\in\mathbb{Z}_k} \psi_i \cdot f_i.$$

This describes the conjunctive metagame obtained by negating f_i whenever $\psi_i = -1$. Negating the function f itself produces the metagame in which MIN attempts to make the result vector equal to ψ. □

Theorem 6.2.6. Let $\Gamma = \langle X^{\mathcal{S}}, f\rangle$ and $\mathcal{S}^* \subseteq \mathcal{S}$. Then $\Gamma * \mathcal{S}^* = \Gamma \wedge \langle X^{\mathcal{S}^*\setminus\mathcal{S}}, +\rangle = \Gamma \vee \langle X^{\mathcal{S}^*\setminus\mathcal{S}}, -\rangle$.

Proof.

$$\Gamma * \mathcal{S}^* \overset{\text{def}}{=\!=} \langle X^{\mathcal{S}^*}, f\rangle = \langle X^{\mathcal{S}\cup(\mathcal{S}^*\setminus\mathcal{S})}, f \wedge \text{TRUE}\rangle = \langle X^{\mathcal{S}}, f\rangle \wedge \langle X^{\mathcal{S}^*\setminus\mathcal{S}}, +\rangle$$

and analogously $\Gamma * \mathcal{S}^* = \Gamma \vee \langle X^{\mathcal{S}^*\setminus\mathcal{S}}, -\rangle$. □

Theorem 6.2.7. Let $\Gamma = \langle\langle\{\Gamma_i\}_{i\in\mathbb{Z}_k}, f\rangle\rangle$ with $\Gamma_i = \langle X^{\mathcal{S}_i}, f_i\rangle$ and $\mathcal{S} = \bigcup_{i\in\mathbb{Z}_k} \mathcal{S}_i$. Let $\psi \in X^{\mathcal{S}}$. Then $\Gamma/\psi = \langle\langle\{\Gamma_i/\psi\}_{i\in\mathbb{Z}_k}, f\rangle\rangle$.

Proof. Without loss of generality assume that $\mathcal{A}(\psi) \subseteq \mathcal{S}$. The colour space of $\langle\langle\{\Gamma_i\}_{i\in\mathbb{Z}_k}, f\rangle\rangle/\psi$ is $\mathcal{S} \setminus \mathcal{A}(\psi) = \big(\bigcup_{i\in\mathbb{Z}_k}(\mathcal{S}_i)\big) \setminus \mathcal{A}(\psi) = \bigcup_{i\in\mathbb{Z}_k}(\mathcal{S}_i \setminus \mathcal{A}(\psi))$ which is the colour space of $\langle\langle\{\Gamma_i/\psi\}_{i\in\mathbb{Z}_k}, f\rangle\rangle$.

For the scoring functions we have

$$[f/\psi](\psi') = f\Big(f_i((\psi'\overline{\psi} \searrow \mathcal{S}) \searrow \mathcal{S}_i)\Big)_{i\in\mathbb{Z}_k} = f\Big(f_i(\psi'\overline{\psi} \searrow \mathcal{S}_i)\Big)_{i\in\mathbb{Z}_k} = f\Big([f_i/\psi](\psi')\Big)_{i\in\mathbb{Z}_k}$$

since $(\psi'\overline{\psi} \searrow \mathcal{S}) \searrow \mathcal{S}_i = \psi'\overline{\psi} \searrow \mathcal{S}_i$ by Observation 2.1.3:iv. □

Theorem 6.3.3. Let $\Gamma = \langle X^{\mathcal{S}}, f\rangle$ and $\mathcal{S}^* \subseteq \mathcal{S}$, then

$$\Gamma * \mathcal{S}^* = \begin{cases} \Gamma & \text{if } |\mathcal{S}^* \setminus \mathcal{S}| \text{ is even,} \\ \Gamma* & \text{if } |\mathcal{S}^* \setminus \mathcal{S}| \text{ is odd.} \end{cases}$$

Corollary i: $(\Gamma_0 * \square) \wedge (\Gamma_1 * \square) = (\Gamma_0 * \triangle) \wedge (\Gamma_1 * \triangle) = \Gamma_0 \wedge \Gamma_1 * \square$, and the same goes for $\vee$ instead of $\wedge$;

Corollary ii: $(\Gamma_0 * \square) \wedge (\Gamma_1 * \triangle) = (\Gamma_0 * \triangle) \wedge (\Gamma_1 * \square) = \Gamma_0 \wedge \Gamma_1 * \triangle$, and the same goes for $\vee$ instead of $\wedge$.

Corollary iii: Let $\Diamond \in \{\square, \triangle\}$ and $\mathfrak{m} \in \mathfrak{M}(\mathcal{S})$, then $(\Gamma * \Diamond) \oplus \mathfrak{m} = (\Gamma \oplus \mathfrak{m}) * \Diamond *$.

Proof. The main theorem is a restatement of Theorem 5.4.2. The corollaries follow in combination with Theorem 5.4.2. □

Theorem 6.3.4. Let $\Gamma = \langle X^{\mathcal{S}}, f\rangle$ and $\Gamma' = \langle X^{\mathcal{S}'}, f'\rangle$ with $\Gamma \leqslant \Gamma'$, and let $\mathfrak{c} \in \mathfrak{C}$. If $|\mathcal{S}\backslash\mathcal{S}'|$ and $|\mathcal{S}'\backslash\mathcal{S}|$ are even then $\textsc{mnx}(\Gamma;\mathfrak{c}) = +1 \Longrightarrow \textsc{mnx}(\Gamma';\mathfrak{c}) = +1$ and $\textsc{mnx}(\Gamma';\mathfrak{c}) = -1 \Longrightarrow \textsc{mnx}(\Gamma;\mathfrak{c}) = -1$. If Γ and Γ' are isotone then the requirement that $|\mathcal{S} \setminus \mathcal{S}'|$ and $|\mathcal{S}' \setminus \mathcal{S}|$ be even can be dropped.

Proof. Let $\mathcal{S}^* = \mathcal{S} \cup \mathcal{S}'$, so that $\Gamma * \mathcal{S}^* \leqslant \Gamma' * \mathcal{S}^*$. If $|\mathcal{S} \setminus \mathcal{S}'|$ and $|\mathcal{S}' \setminus \mathcal{S}|$ are even then $|\mathcal{S}^* \setminus \mathcal{S}|$ and $|\mathcal{S}^* \setminus \mathcal{S}'|$ are also even since $\mathcal{S}^* \setminus \mathcal{S} = \mathcal{S}' \setminus \mathcal{S}$ and $\mathcal{S}^* \setminus \mathcal{S}' = \mathcal{S} \setminus \mathcal{S}'$. We then have

$$\begin{aligned} \textsc{mnx}(\Gamma';\mathfrak{c}) &= \textsc{mnx}(\Gamma' * \mathcal{S}^*;\mathfrak{c}) && \text{(Theorem 5.4.2)} \\ &\geqslant \textsc{mnx}(\Gamma * \mathcal{S}^*;\mathfrak{c}) && (\Gamma * \mathcal{S}^* \leqslant \Gamma' * \mathcal{S}^*) \\ &= \textsc{mnx}(\Gamma;\mathfrak{c}) && \text{(Theorem 5.4.2)} \end{aligned}$$

from which the implications follow. If both games are isotone then by Theorem 5.5.2 the proof also holds when $|\mathcal{S}^* \setminus \mathcal{S}|$ and $|\mathcal{S}^* \setminus \mathcal{S}'|$ are not both even. □

Theorem 6.3.5. Let $\Gamma_0 = \langle X^{\mathcal{S}_0}, f_0\rangle$ and $\Gamma_1 = \langle X^{\mathcal{S}_1}, f_1\rangle$ such that $\Gamma_0 \leqslant \Gamma_1$. For any $\Gamma_2 = \langle X^{\mathcal{S}}, f\rangle$ we then have $\Gamma_0 \wedge \Gamma_2 \leqslant \Gamma_1 \wedge \Gamma_2$ and $\Gamma_0 \vee \Gamma_2 \leqslant \Gamma_1 \vee \Gamma_2$.

Proof. This follows immediately from Observation 6.3.2:i. □

Theorem 6.3.6. Let $\Gamma_i = \langle X^{\mathcal{S}_i}, f_i\rangle$ for $0 \leqslant i \leqslant 2$. If $\Gamma_0 \geqslant \Gamma_2$ and $\Gamma_1 \geqslant \Gamma_2$, then $\Gamma_0 \vee \Gamma_1 \geqslant \Gamma_0 \wedge \Gamma_1 \geqslant \Gamma_2$. If $\Gamma_0 \leqslant \Gamma_2$ and $\Gamma_1 \leqslant \Gamma_2$, then $\Gamma_0 \wedge \Gamma_1 \leqslant \Gamma_0 \vee \Gamma_1 \leqslant \Gamma_2$.

Proof. Since in general $\Gamma_i \geqslant \Gamma_j$ is equivalent to $\forall_{\psi^* \in \check{X}^{\mathcal{S}_i \cup \mathcal{S}_j}} [f_i(\psi^*) \Longleftarrow f_j(\psi^*)]$, the theorem is a rewrite of the elementary Boolean equations

$$(t_0 \wedge t_1) \Longrightarrow (t_0 \vee t_1),$$
$$\big((t_2 \Longrightarrow t_0) \wedge (t_2 \Longrightarrow t_1)\big) \Longleftrightarrow \big(t_2 \Longrightarrow (t_0 \wedge t_1)\big),$$
$$\big((t_0 \Longrightarrow t_2) \wedge (t_1 \Longrightarrow t_2)\big) \Longleftrightarrow \big((t_0 \vee t_1) \Longrightarrow t_2\big).$$
□

Theorem 6.3.7. Let $\mathcal{Q} = \{\Gamma_i\}_{i \in \mathbb{Z}_k}$. Put $\mathcal{Q}' = \{\Gamma_0*, \Gamma_1, \Gamma_2, \ldots, \Gamma_{k-1}\}$. Then $\langle\langle \mathcal{Q}', f\rangle\rangle = \langle\langle \mathcal{Q}, f\rangle\rangle*$.

Proof. First consider the base cases. Using Theorem 6.2.6 we have:

$$(\Gamma_0*) \wedge \Gamma_1 = (\Gamma_0 \wedge \langle X^w, +\rangle) \wedge \Gamma_1 = (\Gamma_0 \wedge \Gamma_1) \wedge \langle X^w, +\rangle = (\Gamma_0 \wedge \Gamma_1)*;$$
$$(\Gamma_0*) \vee \Gamma_1 = (\Gamma_0 \vee \langle X^w, -\rangle) \vee \Gamma_1 = (\Gamma_0 \vee \Gamma_1) \vee \langle X^w, -\rangle = (\Gamma_0 \wedge \Gamma_1)*;$$
$$-(\Gamma_0*) = -(\Gamma_0 \wedge \langle X^w, +\rangle) = (-\Gamma_0) \vee (-\langle X^w, +\rangle) = (-\Gamma_0) \vee \langle X^w, -\rangle = (-\Gamma_0)*$$

where w is the dead element and negating a game means negating its scoring function. Since any metafunction can be thus composed of conjunction, disjunction, and negation, the lemma holds. □

Theorem 6.4.1. Let $\mathcal{Q} = \{\langle X^{\mathcal{S}_i}, f_i\rangle\}_{i \in \mathbb{Z}_k}$ be a partition game with $\mathcal{S} = \bigcup_{i \in \mathbb{Z}_k} \mathcal{S}_i$. Then

$$\langle\langle \mathcal{Q}, \vee\rangle\rangle \;\leqslant\; \Gamma_i \;\leqslant\; \langle\langle \mathcal{Q}, \wedge\rangle\rangle$$

for all $i \in \mathbb{Z}_k$.

Corollary i: For any $\mathfrak{c} \in \mathfrak{C}$, if $\exists_{i \in \mathbb{Z}_k}\big[\text{MNX}(\Gamma_i * \mathcal{S}; \mathfrak{c}) = +1\big]$ then $\text{MNX}\big(\langle\langle \mathcal{Q}, \vee\rangle\rangle; \mathfrak{c}\big) = +1$.

Corollary ii: For any $\mathfrak{c} \in \mathfrak{C}$, if $\exists_{i \in \mathbb{Z}_k}\big[\text{MNX}(\Gamma_i * \mathcal{S}; \mathfrak{c}) = -1\big]$ then $\text{MNX}\big(\langle\langle \mathcal{Q}, \wedge\rangle\rangle; \mathfrak{c}\big) = -1$.

Proof. Let $\Gamma_i' = \bigvee_{j \in \mathbb{Z}_k \setminus i} \Gamma_i$, so that $\langle\langle \mathcal{Q}, \vee\rangle\rangle = \Gamma_i \vee \Gamma_i'$. By Theorems 6.2.6 and 6.3.5 we then have $\Gamma_i * \mathcal{S} = \Gamma_i \vee \langle X^{\mathcal{S}\setminus\mathcal{S}_i}, -\rangle \geqslant \Gamma_i \vee \Gamma_i' = \langle\langle \mathcal{Q}, \vee\rangle\rangle$. Analogously, $\Gamma_i * \mathcal{S} = \Gamma_i \wedge \langle X^{\mathcal{S}\setminus\mathcal{S}_i}, +\rangle \leqslant \langle\langle \mathcal{Q}, \wedge\rangle\rangle$. The corollaries follow from Observations 6.3.1:i and 6.3.2:iv. □

Theorem 6.4.2. Let $\mathcal{Q} = \{\langle X^{\mathcal{S}_i}, f_i\rangle\}_{i \in \mathbb{Z}_k}$ be a partition game with $\mathcal{S} = \bigcup_{i \in \mathbb{Z}_k} \mathcal{S}_i$. Then:

$$\forall_{i \in \mathbb{Z}_k}\big[\text{MNX}(\Gamma_i * \Box; \text{MIN}) = +1\big] \Longrightarrow \text{MNX}\big(\langle\langle \mathcal{Q}, \wedge\rangle\rangle * \Box; \text{MIN}\big) = +1;$$
$$\forall_{i \in \mathbb{Z}_k}\big[\text{MNX}(\Gamma_i * \Box; \text{MAX}) = -1\big] \Longrightarrow \text{MNX}\big(\langle\langle \mathcal{Q}, \vee\rangle\rangle * \Box; \text{MAX}\big) = -1;$$

Proof of main theorem. First consider the case with two even components, so that $k = 2$, $\Gamma_0 * \Box = \Gamma_0$, $\Gamma_1 * \Box = \Gamma_1$, and $(\Gamma_0 \wedge \Gamma_1) * \Box = \Gamma_0 \wedge \Gamma_1$. The proof for this case goes by induction to $|\mathcal{S}_0|$ and $|\mathcal{S}_1|$.

Base case: $|\mathcal{S}_0| = |\mathcal{S}_1| = 0$. Then $\Gamma_0 = \langle \varnothing, t_0\rangle$, $\Gamma_1 = \langle \varnothing, t_1\rangle$, and $\langle\langle \mathcal{Q}, \wedge\rangle\rangle = \langle \varnothing, t_0 \wedge t_1\rangle$. Since for both components we have $\text{MNX}(\Gamma_i * \mathcal{S}; \text{MIN}) = +1$ and $\Gamma_i * \mathcal{S} = \Gamma_i$, so that apparently $t_0 = t_1 = +1$ and therefore $\text{MNX}(\langle\langle \mathcal{Q}, \wedge\rangle\rangle; \text{MIN}) = t_0 \wedge t_1 = +1$.

Induction step. Let $\mathfrak{p} \in \mathfrak{P}_{\mathcal{S}}$ such that $\mathfrak{p} \leftarrow (\phi^{\mathcal{S}}, \text{MIN})$. Then $\mathfrak{p} = (\psi, \text{MAX})$ for some $\psi \in X^{\mathcal{S}}$, and there is a $\mathfrak{m} \in \mathfrak{M}(\phi^{\mathcal{S}})$ such that $\mathfrak{p} = (\phi^{\mathcal{S}}, \text{MIN}) \oplus \mathfrak{m}$ and $\psi = \phi^{\mathcal{S}}\mathfrak{m}$. Since $\mathcal{S} = \mathcal{S}_0 \cup \mathcal{S}_1$ and $\mathcal{S}_0 \cap \mathcal{S}_1 = \varnothing$, without loss of generality assume that $\mathfrak{m} \in \mathfrak{M}(\phi^{\mathcal{S}_0})$. Put $\mathfrak{p}_0 = \mathfrak{p} \searrow \mathcal{S}_0$, then $\mathfrak{p}_0 = (\phi^{\mathcal{S}}\mathfrak{m}, \text{MAX}) \searrow \mathcal{S}_0 = (\phi^{\mathcal{S}}\mathfrak{m} \searrow \mathcal{S}_0, \text{MAX}) = (\phi^{\mathcal{S}_0}\mathfrak{m}, \text{MAX}) \leftarrow (\phi^{\mathcal{S}_0}, \text{MIN})$. The component Γ_0 is even, so $\text{MNX}(\Gamma_0 *; \text{MIN}) = +1$ implies that $\text{MNX}(\Gamma_0; \text{MIN}) = +1$ according to Theorem 6.3.3. This means that $\text{NGX}(\Gamma_0; \text{MIN}) = -1$, so $\text{NGX}(\Gamma_0; \mathfrak{p}_0) = +1$ from Definition 5.1.3, and from the same definition we have the existence of $\mathfrak{m}' \in \mathfrak{M}(\phi^{\mathcal{S}_0}\mathfrak{m})$ such that $\text{NGX}(\Gamma_0; \mathfrak{p}_0 \oplus \mathfrak{m}') = -1$. Since $\mathfrak{p}_0 \oplus \mathfrak{m}' = (\phi^{\mathcal{S}_0}\mathfrak{m}\mathfrak{m}', \text{MIN})$ we have $\text{NGX}(\Gamma_0/\mathfrak{m}\mathfrak{m}'; \text{MIN}) = -1$ and thus $\text{MNX}(\Gamma_0/\mathfrak{m}\mathfrak{m}'; \text{MIN}) = +1$. Both $\mathfrak{m}$ and $\mathfrak{m}'$ are not moves in Γ_1 because $\mathcal{A}(\mathfrak{m}), \mathcal{A}(\mathfrak{m}') \in \mathcal{S}_0$ and $\mathcal{S}_0 \cap \mathcal{S}_1 = \varnothing$, so that $\Gamma_1/\mathfrak{m}\mathfrak{m}' = \Gamma_1$ from Observation 6.1.1:ii. Now both $\Gamma_0/\mathfrak{m}\mathfrak{m}'$ and $\Gamma_1/\mathfrak{m}\mathfrak{m}'$ are even, and $\Gamma_0/\mathfrak{m}\mathfrak{m}'$ is of lower dimension than Γ_0, so by induction we have $\text{MNX}(\Gamma_0 \wedge \Gamma_1; \mathfrak{p} \oplus \mathfrak{m}') = \text{MNX}(\Gamma_0 \wedge \Gamma_1; (\phi^{\mathcal{S}}, \text{MIN}) \oplus \mathfrak{m} \oplus \mathfrak{m}') = \text{MNX}\big((\Gamma_0 \wedge \Gamma_1)/\mathfrak{m}\mathfrak{m}'; \text{MIN}\big) = +1$. This holds for any $\mathfrak{p} \leftarrow (\phi^{\mathcal{S}}, \text{MIN})$, and therefore $\text{MNX}(\Gamma_0 \wedge \Gamma_1; \phi^{\mathcal{S}}, \text{MIN}) = +1$.

This then applies equally well to the general case for $k = 2$ where one or both components may be even, as both $\Gamma_0 * \square$ and $\Gamma_1 * \square$ are even, and $(\Gamma_0 * \square) \wedge (\Gamma_1 * \square) = (\Gamma_0 \wedge \Gamma_1) * \square$ according to Theorem 6.3.3.

The proof for the general case where $k > 2$ goes by induction to the number of components, treating $\Gamma_1 \wedge \Gamma_2 \wedge \ldots \wedge \Gamma_{k-1}$ as one component. This proves the case for $\text{MNX}(\langle\langle \mathcal{Q}, \wedge \rangle\rangle; \text{MIN})$; the proof for $\text{MNX}(\langle\langle \mathcal{Q}, \vee \rangle\rangle; \text{MAX})$ is identical with the roles of MIN and MAX interchanged. □

Theorem 6.4.3. Let $\mathcal{Q} = \{\langle X^{\mathcal{S}_i}, f_i \rangle\}_{i \in \mathbb{Z}_k}$ be a family of independent games, and let $\Diamond \in \{\square, \triangle\}$. If there exists a Γ_j such that $\text{MNX}(\Gamma_j * \Diamond; \text{MIN}) = -1$, then any $\mathfrak{m} \notin \mathfrak{M}(\mathcal{S}_j)$ is a losing move in $(\langle\langle \mathcal{Q}, \wedge \rangle\rangle * \Diamond *, \text{MAX})$. If there exists a Γ_j such that $\text{MNX}(\Gamma_j * \Diamond; \text{MAX}) = +1$, then any $\mathfrak{m} \notin \mathfrak{M}(\mathcal{S}_j)$ is a losing move in $(\langle\langle \mathcal{Q}, \vee \rangle\rangle * \Diamond *, \text{MIN})$.

> *Corollary i:* Let Γ_i and Γ_j be two components such that $i \neq j$. If $\text{MNX}(\Gamma_i * \Diamond; \text{MIN}) = -1$ and $\text{MNX}(\Gamma_j * \Diamond; \text{MIN}) = -1$, then $\text{MNX}(\langle\langle \mathcal{Q}, \wedge \rangle\rangle * \Diamond *; \text{MAX}) = -1$. If $\text{MNX}(\Gamma_i * \Diamond; \text{MAX}) = +1$ and $\text{MNX}(\Gamma_j * \Diamond; \text{MAX}) = +1$, then $\text{MNX}(\langle\langle \mathcal{Q}, \vee \rangle\rangle * \Diamond *; \text{MAX}) = +1$.

Proof. First consider the case with $\text{MNX}(\Gamma_j * \Diamond; \text{MIN}) = -1$. Put $\Gamma = \langle\langle \mathcal{Q}, \wedge \rangle\rangle$, and without loss of generality let $j \neq 0$ and let $\mathfrak{m} \in \mathfrak{M}(\Gamma_0)$. Then $\big(\Gamma * \Diamond *\big) \oplus \mathfrak{m} = (\Gamma \oplus \mathfrak{m}) * \Diamond$ according to Theorem 6.3.3. This game has parity $\Diamond$ and contains the component $\Gamma_j/\mathfrak{m} = \Gamma_j$ since $\mathfrak{m} \notin \mathfrak{M}(\mathcal{S}_j)$. This fulfills the preconditions of the corollaries of Theorem 6.4.1, implying the conclusion. The proof for the case $\text{MNX}(\Gamma_j * \Diamond; \text{MAX}) = +1$ is analogous. For the corollary note that for any move $\mathfrak{m}$ we must have either $\mathfrak{m} \notin \mathfrak{M}(\Gamma_i)$ or $\mathfrak{m} \notin \mathfrak{M}(\Gamma_j)$ since $\mathcal{S}_i \cap \mathcal{S}_j = \varnothing$, so that $\mathfrak{m}$ is a losing move. □

Theorem 6.4.4. Let $\mathcal{Q} = \{\langle X^{\mathcal{S}_i}, f_i \rangle\}_{i \in \mathbb{Z}_k}$ be set of independent games. If there are no empty components, then:

$$\exists_{i \in \mathbb{Z}_k}\big[\text{MNX}(\Gamma_i * \triangle; \text{MAX}) = +1\big] \wedge \forall_{j \in \mathbb{Z}_k \setminus i}\big[\text{MNX}(\Gamma_j * \square; \text{MIN}) = +1\big] \Longrightarrow \text{MNX}\big(\langle\langle \mathcal{Q}, \wedge \rangle\rangle * \triangle; \text{MAX}\big) = +1;$$

$$\exists_{i \in \mathbb{Z}_k}\big[\text{MNX}(\Gamma_i * \triangle; \text{MIN}) = -1\big] \wedge \forall_{j \in \mathbb{Z}_k \setminus i}\big[\text{MNX}(\Gamma_j * \square; \text{MAX}) = -1\big] \Longrightarrow \text{MNX}\big(\langle\langle \mathcal{Q}, \vee \rangle\rangle * \triangle; \text{MIN}\big) = -1.$$

Proof. Without loss of generality let Γ_0 be the component such that $\text{MNX}(\Gamma_0 * \triangle; \text{MAX}) = +1$, and so $\text{MNX}(\Gamma_i * \square; \text{MIN}) = +1$ for all $1 \leqslant i < k$. If $k = 1$ then $\langle\langle \mathcal{Q}, \wedge \rangle\rangle = \Gamma_0$ and there is nothing to prove, so assume $k > 1$. Since $\text{MNX}(\Gamma_0 * \triangle; \text{MAX}) = +1$ there exists $\mathfrak{m} \in \mathfrak{M}(\phi^{\mathcal{S}_0 * \triangle})$ such that $\text{MNX}((\Gamma_0 * \triangle)/\mathfrak{m}; \text{MIN}) = \text{MNX}(\Gamma_0; \phi^{\mathcal{S}_0 * \triangle}\mathfrak{m}, \text{MIN}) = \text{MNX}(\Gamma_0/\mathfrak{m}; \phi^{\mathcal{S}_0 * \square}, \text{MIN}) = \text{MNX}((\Gamma_0/\mathfrak{m}) * \square; \text{MIN}) = -1$. For $1 \leqslant i < k$ we have $\mathcal{A}(\mathfrak{m}) \notin \mathcal{S}_i$ because the components are independent, so $\Gamma_i/\mathfrak{m} = \Gamma_i$. Therefore $\text{MNX}(\Gamma_i/\mathfrak{m} * \square; \text{MIN}) = \text{MNX}(\Gamma_i * \square; \text{MIN}) = -1$. Theorem 6.4.2 then implies that $\text{MNX}((\langle\langle \mathcal{Q}, \wedge \rangle\rangle * \square)/\mathfrak{m}; \text{MIN}) = -1$. Therefore $\text{MNX}(\langle\langle \mathcal{Q}, \wedge \rangle\rangle * \triangle; \text{MAX}) = +1$. The proof for the implication for $\text{MNX}(\langle\langle \mathcal{Q}, \vee \rangle\rangle; \text{MIN})$ is identical, with the roles of MIN and MAX interchanged. □

Theorem 6.5.1. Let $\mathcal{Q} = \{\Gamma_i\}_{i \in \mathbb{Z}_k}$ be a family of isotone pairwise independent games. Let n_{P}, n_{R}, and n_{N} be the number of positive, regular, and negative components.

$$\langle\langle \mathcal{Q}, \wedge \rangle\rangle \quad \text{is} \quad \begin{cases} \text{positive} & \text{if } n_{\text{R}} = 0 \text{ and } n_{\text{N}} = 0; \\ \text{regular} & \text{if } n_{\text{R}} = 1 \text{ and } n_{\text{N}} = 0; \\ \text{negative} & \text{if } n_{\text{R}} \geqslant 2 \text{ or } n_{\text{N}} \geqslant 1. \end{cases}$$

$$\langle\langle \mathcal{Q}, \vee \rangle\rangle \quad \text{is} \quad \begin{cases} \text{negative} & \text{if } n_{\text{R}} = 0 \text{ and } n_{\text{P}} = 0; \\ \text{regular} & \text{if } n_{\text{R}} = 1 \text{ and } n_{\text{P}} = 0; \\ \text{positive} & \text{if } n_{\text{R}} \geqslant 2 \text{ or } n_{\text{P}} \geqslant 1. \end{cases}$$

Proof. Put $\Gamma_i = \langle X^{\mathcal{S}_i}, f_i \rangle$ and $\mathcal{S} \supseteq \bigcup_{i \in \mathbb{Z}_k} \mathcal{S}_i$ such that $|\mathcal{S}|$ is even. For every $\Gamma_i \in \mathcal{Q}$ we have $\text{MNX}(\Gamma_i * \mathcal{S}; \mathfrak{c}) = \text{MNX}(\Gamma_i; \mathfrak{c})$ by Theorem 5.5.2 and Theorem 6.3.3 because Γ_i is isotone, and for the same reason $\text{MNX}(\langle\langle \mathcal{Q}, \wedge \rangle\rangle; \mathfrak{c}) = \text{MNX}(\langle\langle \mathcal{Q}, \wedge \rangle\rangle * \mathcal{S}; \mathfrak{c})$.

Case 1: $n_{\text{R}} = 0$ and $n_{\text{N}} = 0$. Then each component Γ_i is positive, so $\text{MNX}(\Gamma_i; \text{MIN}) = +1$ and therefore $\text{MNX}(\Gamma_i * \mathcal{S}; \text{MIN}) = +1$, from which Theorem 6.4.2 implies that $\text{MNX}(\langle\langle \mathcal{Q}, \wedge \rangle\rangle; \text{MIN}) = +1$. Since $\langle\langle \mathcal{Q}, \wedge \rangle\rangle$ is isotone that also implies that $\text{MNX}(\langle\langle \mathcal{Q}, \wedge \rangle\rangle; \text{MAX}) = +1$ by Corollary 5.5.3:iii. Therefore $\langle\langle \mathcal{Q}, \wedge \rangle\rangle$ is positive.

Case 2: $n_{\text{R}} = 1$ and $n_{\text{N}} = 0$. Without loss of generality let Γ_0 be the regular component. Since $\text{MNX}(\Gamma_0; \text{MIN}) = -1$ we have $\text{MNX}(\Gamma_0 * \mathcal{S}; \text{MIN}) = -1$ and therefore $\text{MNX}(\langle\langle \mathcal{Q}, \wedge \rangle\rangle; \text{MIN}) = -1$ by Theorem 6.4.2. Since $\text{MNX}(\Gamma_0; \text{MAX}) = +1$ there is a move χ^v such that $\text{MNX}(\Gamma_0/\chi^v; \text{MIN}) = \text{MAX}$. Then Γ_0/χ^v and all the remaining components are positive, from which Case 1 implies that $\text{MNX}(\langle\langle \mathcal{Q}, \wedge \rangle\rangle/\chi^v; \text{MIN}) = \text{MAX}$, and therefore $\text{MNX}(\langle\langle \mathcal{Q}, \wedge \rangle\rangle; \text{MAX}) = \text{MAX}$. Thus $\langle\langle \mathcal{Q}, \wedge \rangle\rangle$ is regular.

Case 3: $n_{\text{R}} \geqslant 2$. Without loss of generality let Γ_0 and Γ_1 be negative. For any move χ^v, at least one of Γ_0 and Γ_1 does not contain v and is still regular. Therefore by Theorem 6.4.1 we have $\text{MNX}(\langle\langle \mathcal{Q}, \wedge \rangle\rangle/\chi^v; \text{MIN}) = -1$, and thus $\text{MNX}(\langle\langle \mathcal{Q}, \wedge \rangle\rangle; \text{MAX}) = -1$. Since $\langle\langle \mathcal{Q}, \wedge \rangle\rangle$ is isotone, it is negative.

Case 4: $n_{\text{N}} \geqslant 1$. Without loss of generality let Γ_0 be the negative component. By Theorem 6.4.1 we have $\text{MNX}(\langle\langle \mathcal{Q}, \wedge \rangle\rangle; \text{MAX}) = -1$. As in Case 3, this implies that $\langle\langle \mathcal{Q}, \wedge \rangle\rangle$ is negative.

The proofs for $\langle\langle \mathcal{Q}, \vee \rangle\rangle$ are entirely analogous, with interchanged roles for MIN and MAX. □

Chapter 7

Combinatorial Game Theory

Section 6.4 provided some theorems about determining the minimax value of a metagame by independent analysis of the components. The theorems are correlated to strategy advice where the winning player employs a partition strategy, and involve cases where the winning player needs to win only one component, or makes the last move of the game. When this is not the case, it seems that any winning strategy must be more subtle, dealing with intricate interplay between components.

The field of **Combinatorial Game Theory** (CGT) has been created for just such purposes [23, 13]. CGT methods can be used and modified to shed more light on the behaviour of partition games. The discussion in this chapter also appeared in [85]; the developed theory applies not only to set colouring games but to a more general class of "binary combinatorial games", the difference being that a binary combinatorial game need not have a fixed length in terms of number of moves played. The notation and terminology in this chapter follows common CGT conventions and assumes familiarity with such conventions; refer to [13] for the standard introduction to these topics.

7.1 Binary Combinatorial Values

In order to introduce binary combinatorial values, the **division game** will be more useful. In this game the two players take turns acquiring one of the variables, and when all variables have been claimed a function assigns the win to one of the players. This is equivalent to GAME-SAT with the additional restriction that MAX may only assign the value TRUE to a variable, and MIN may only assign FALSE. Division games were introduced by Yamasaki [102], who also extended them to games with a predetermined but not necessarily alternating order of play; in this chapter, only alternating play will be assumed.

The notation for binary games will mirror the standard CGT conventions, with MAX playing the role of Left and MIN playing Right. Where combinatorial games are built up starting with the "atomic" game $\{\,|\,\}$, binary games start with the atomic games T and F.

Definition 7.1.1. A **binary game** is a game that has no stopping positions other than T and F.

This precludes "empty" stopping positions, so that games like $\{\text{T}|\}$ are not allowed.

If G is some binary game, then its negation $\overline{G}$ is obtained essentially by MIN and MAX switching roles. This is the binary counterpart of the negative of a game in regular CGT, so we have:

Definition 7.1.2. $\overline{G} = \{\overline{G^R}|\overline{G^L}\}$.

For the base cases we have of course $\overline{\text{T}} = \text{F}$ and $\overline{\text{F}} = \text{T}$. Indeed it will be supposed from now on that any assertion involving only T and F is to behave "as expected".

The common convention in CGT is "normal play", where the first player unable to move is the loser. This is not the goal in set colouring games, where in fact the first player unable to move is determined from the start of the game no matter how play proceeds. To turn a binary game into a combinatorial game, allow the winner one more move at the end of the game. Then the game T obtains the value $+1$, and the game F has the value -1. Indeed it does not really matter exactly how many extra moves are awarded at the end of the game, so T and F could be represented by any combination of a positive and a negative integer. If x is the value of a binary game G, then the value of $\overline{G}$ is $-x$ provided that we choose $\text{F} = -\text{T}$.

Definition 7.1.3. The combinatorial value of a binary game is obtained by replacing all T's with some positive integer k, and replacing all F's with $-k$.

In CGT, a game in which both players always have the same moves available is called an **impartial game**. It has been shown that every impartial game is essentially a variant of the game of Nim [13]. The CGT analysis of set colouring games makes them *almost* impartial: both players have the same options available in all positions, but the game ends in the asymmetrical value T or F instead of 0.

7.2 Conjunctions and Disjunctions

Binary games combine not by taking their sum, but instead by taking their conjunction or disjunction. If the players do calculate the sum $G + H$, they will find that the sum is positive when MAX can win $G \wedge H$, and negative if MIN can win $G \vee H$. But these conditions are not sufficient, as it is not yet clear what happens when both players can win one component. The status of the sum would then depend on who gets the last move, and when fighting over the last move the players are no longer playing the same game as the conjunction or disjunction.[1]

The solution is to make the players play the combination of *three* binary games, avoiding the possibility of a "draw". To win the sum of three binary games, one needs to win at least two of the

[1] Note that the definition of binary games does not specify that they have a fixed length, so it is possible for fights to arise over the right to move last.

components. We then have:

Theorem 7.2.1. *Let G and H be binary games. The result of $G \wedge H$ is equal to the result of $G + H + \mathrm{F}$. The result of $G \vee H$ is equal to the result of $G + H + \mathrm{T}$.*

In effect, MIN is given an extra move in $G \wedge H$, to counteract MAX's win in one component. Crucially, this extra move does not spoil the legality of play in the binary game $G \wedge H$, because by the Number Avoidance Theorem[2] the result is unaltered when MIN is prohibited from using this extra move until the other components have settled on an integer as well.

Continuing on this, it becomes evident that the result of $G \wedge H \wedge K$ is equal to the result of $G + H + K + 2\mathrm{F}$ and so on, so that the outcome of the conjunction or disjunction of k components can be determined by comparing the sum to $(k-1)\mathrm{T}$ or $(k-1)\mathrm{F}$, respectively.

The expressions $G > 0$, $G < 0$, $G \| 0$, and $G = 0$ will be used in the same way as for combinatorial games, indicating wins for MAX, MIN, the first player, and the second player. The "Tweedle-Dee and Tweedle-Dum argument" then shows that $G \wedge \overline{G} \leqslant 0$ and $G \vee \overline{G} \geqslant 0$, because the second player can ensure that the two components end in opposite outcomes by always copying the opponent's move in the other component. Theorem 7.2.1 actually strengthens this to $G \wedge \overline{G} < 0$ and $G \vee \overline{G} > 0$.

7.3 Order Relation

The order relation $\geqslant$ for binary games is defined by comparing the combinatorial values.

Definition 7.3.1. *Let G and H be binary games. Then $G \geqslant H$ if and only if the same relation holds between their combinatorial values.*

There is a binary games analogue to testing $G \geqslant H$ by playing their combinatorial difference. If $G \geqslant H$ then $G - H \geqslant 0$ so $G - H + \mathrm{T} > 0$, which means that $G \vee \overline{H}$ must also be positive. Conversely, if $G \lhd\!| H$ then $G - H \lhd\!| 0$ so $G - H + \mathrm{F} \lhd\!| 0$, thus if $G \wedge \overline{H} \geqslant 0$ then $G \geqslant H$.

Theorem 7.3.2. *For $G \geqslant H$ it is necessary that $G \vee \overline{H} > 0$ and it is sufficient that $G \wedge \overline{H} \geqslant 0$.*

Unfortunately there usually is a "gap" between these conditions, so the comparison between G and H cannot always be resolved by playing a combination of G and $\overline{H}$. It can be resolved, of course, by comparing the combinatorial values of G and H.

The statement "$G \geqslant H$ if no $G^R \leqslant H$ and $G \leqslant$ no H^L" does not work for binary games, as it would get off on the wrong foot right away with $\mathrm{T} \geqslant \mathrm{F} \geqslant \mathrm{T}$, since neither T nor F has any options at all. However, it does hold for all the values to be encountered in this chapter, provided that no atomic games are used. When replacing T and F with the later to be encountered expressions $\{\mathrm{T}* | \mathrm{T}*\}$ and $\{\mathrm{F}* | \mathrm{F}*\}$ it works for those values as well, but then it becomes a "bootstrap definition" as T$*$ and F$*$

[2] See [13].

are themselves defined in terms of T and F, so this can not be used for a recursive definition of $\geqslant$.

The intuitive meaning of $G \geqslant H$ in combinatorial games is that Left always prefers G over H, even as part of a sum, irrespective of the other summands. The same is true for conjunctions and disjunctions. Let $G \geqslant H$, and let K be some other binary game. If $H \wedge K \geqslant 0$ then $G + K + \mathrm{F} \geqslant H + K + \mathrm{F} \geqslant 0$ so $G \wedge K \geqslant 0$. Similarly, if $H \wedge K \mathrel{|\!\rhd} 0$ then $G + K + \mathrm{F} \geqslant H + K + \mathrm{F} \mathrel{|\!\rhd} 0$ so $G \wedge K \mathrel{|\!\rhd} 0$. The same holds with $\vee$ in place of $\wedge$.

Theorem 7.3.3. If $G \geqslant H$ then MAX prefers G over H in any conjunction or disjunction.

The important consequence is that if two binary games have the same combinatorial value, then one can be substituted for the other in any conjunction or disjunction. This enables the use of canonical forms for binary games, which is the motivation for Definition 7.3.1.

7.4 Canonical Forms

The formulas of Theorem 7.2.1 reveal the outcome of a conjunction or disjunction, but do not always give its correct value, for otherwise the games $G \wedge (H \vee K)$ and $G \vee (H \wedge K)$ would have the same value $G + H + K$. The root cause is the fact that $\mathrm{T} \vee \mathrm{T} = \mathrm{T}$ and $\mathrm{F} \wedge \mathrm{F} = \mathrm{F}$, where the formulas of Theorem 7.2.1 would give 3T and 3F.

The value of a conjunction or disjunction can be found recursively in the same way as used for sums:

Definition 7.4.1. Let G and H be binary games. Their conjunction $G \wedge H$ and disjunction $G \vee H$ are:

$$G \wedge H = \{G^L \wedge H,\ G \wedge H^L \mid G^R \wedge H,\ G \wedge H^R\},$$
$$G \vee H = \{G^L \vee H,\ G \vee H^L \mid G^R \vee H,\ G \vee H^R\}.$$

It can be verified readily that conjunctions and disjunctions are associative and distributive.

In the theory of surreal numbers, care was taken not to define xy as $\{x^L y, xy^L | x^R y, xy^R\}$, since this would end up defining the same function as $x+y$. However, the definitions for $\wedge$ and $\vee$ do not define the same function, since they behave differently for the atomic games T and F. Strictly speaking only one of these two operators is needed, as it can be verified that indeed $\overline{G} \vee \overline{H} = \overline{G \wedge H}$ and $\overline{G} \wedge \overline{H} = \overline{G \vee H}$, but it seems unfair to choose one as more "basic" than the other.

The same construction can be used to define the game "$G \Rightarrow H$", and by induction it then turns out that this is the same as $\overline{G} \vee H$. This takes care of all the Boolean combinations of two games, except $G \equiv H$ and $\overline{G} \equiv H$, where '$\equiv$' is binary equivalence and the latter game is the exclusive-or. These will be ignored for now, as they turn out to exhibit different and more complicated behaviour.

When one of the games is atomic then it has no left or right options, so we get $G \wedge \mathrm{T} = \{G^L \wedge$

$\mathrm{T} \mid G^R \wedge \mathrm{T}\}$ and so on. Rigorously following the definition eventually leads to the base cases $\{\mathrm{T}|\mathrm{T}\}$, $\{\mathrm{F}|\mathrm{F}\}$, $\{\mathrm{T}|\mathrm{F}\}$ and $\{\mathrm{F}|\mathrm{T}\}$. These have the property that all of their stopping positions are the same. In general, call a game all-TRUE if all its stopping positions are TRUE, and all-FALSE if they are all FALSE.

Consider the conjunction $G \wedge \{\mathrm{F}|\mathrm{T}\}$, where G is all-TRUE. Neither player wants to move in $\{\mathrm{F}|\mathrm{T}\}$, since that would immediately decide the whole game in favour of the opponent. The game $G \wedge \{\mathrm{F}|\mathrm{T}\}$ is therefore decided by a fight for the last move in G, so it behaves like a regular combinatorial game. The combinatorial value of an all-TRUE game G is infinitesimally close to T, because the combinatorial value $G - \mathrm{T}$ is all-small. The notation for all-TRUE and all-FALSE games will mirror the regular CGT notations, so for example $\{\mathrm{T}|\mathrm{T}\} = \mathrm{T}*$ and $\{\mathrm{T}||\mathrm{T}|\mathrm{T}\} = \mathrm{T}\uparrow$.

According to the definition of $\wedge$, the game $G \wedge \mathrm{F}$ behaves exactly like G with all stopping positions replaced by F. This produces an all-FALSE game, infinitesimally close to F. Similarly, $G \vee \mathrm{T}$ is an all-TRUE game. For the cases $G \wedge \mathrm{T}$ and $G \vee \mathrm{F}$ it is easily seen by induction that $G \wedge \mathrm{T} = G \vee \mathrm{F} = G$. This is the binary games analogue of the combinatorial games identity $G + 0 = G$.

7.5 Parity

Adding a dead variable to a set colouring game has the same effect as adding a star to a combinatorial game. Both represent one extra "skip" move, to be used by any player. The combinatorial value of $G*$ is obtained by adding a star to the combinatorial value of G, for we have

$$G* = \{G^L*, G|G^R*, G\}$$

where G^L and G^R are the typical left and right options of G.

In a conjunction or disjunction of games, it does not matter which component a dead variable belongs to, or that it belongs to any component at all:

Theorem 7.5.1. Let G and H be binary games, then $(G*)\wedge H = (G\wedge H)*$ and $(G*)\vee H = (G\vee H)*$.

This follows by induction. Another way of seeing this is to note that $G* = G \wedge (\mathrm{T}*)$, so that $(G*) \wedge H = G \wedge (\mathrm{T}*) \wedge H = (G \wedge H) \wedge (\mathrm{T}*) = (G \wedge H)*$. For the disjunction, use $G* = G \vee (\mathrm{F}*)$.

A GAME-SAT or division game instance has the property that the identity of the last player to move depends only on the parity of the number of variables to be assigned. Call a binary game **even** or **odd** if it always ends after an even or odd number of moves, respectively. Then for even games we have $G \vee \mathrm{T} = \mathrm{T}$ and $G \wedge \mathrm{F} = \mathrm{F}$, and for odd games we have $G \vee \mathrm{T} = \mathrm{T}*$ and $G \wedge \mathrm{F} = \mathrm{F}*$.

A game like $\mathrm{T}\uparrow$ is neither odd nor even. But it can easily be made even, by adding a star to every stop that occurs after an odd number of moves. Similarly a game can be made odd. To be precise:

Definition 7.5.2. If G has options, then $G*\square \overset{\text{def}}{=\!=} \{G^L*\triangle|G^R*\triangle\}$ and $G*\triangle \overset{\text{def}}{=\!=} \{G^L*\square|G^R*\square\}$. If G has no options then $G*\square \overset{\text{def}}{=\!=} G$ and $G*\triangle \overset{\text{def}}{=\!=} G*$.

The games G and $\overline{G}$ have the same parity. A conjunction or disjunction of two binary games is even if and only if the components are both even or both odd.

7.6 Decomposable Games

Consider the previously encountered games $\{\mathrm{T}|\mathrm{F}\}$ and $\{\mathrm{F}|\mathrm{T}\}$. Denote $\mathrm{X} = \{\mathrm{T}|\mathrm{F}\}$, so called because it corresponds to a division game played on just one variable, "$f(x) = x$". Its value is the switch $\pm\mathrm{T}$, revealing $\mathrm{F}* < \mathrm{X} < \mathrm{T}*$ and $\mathrm{F}||\mathrm{X}||\mathrm{T}$. The game $\{\mathrm{F}|\mathrm{T}\}$ corresponds to a division game involving one single variable that neither player wants to own. It does not occur as a one-variable GAME-SAT instance. This game shall be denoted as $\mathrm{Z}_\triangle$, indicating that it is the "odd zero". We obtain $\mathrm{F} < \mathrm{Z}_\triangle < \mathrm{F}$ as well as $\mathrm{F}* < \mathrm{Z}_\triangle < \mathrm{T}*$ and $\mathrm{Z}_\triangle||\mathrm{X}$.

In regular CGT, when $G \geqslant 0$, then Left can win when Right has the first move. The reason is that $G \geqslant 0$ means "no $G^R \leqslant 0$", which by recursion means that no move by Right leads to a game that Right wins when Left moves next, so all moves for Right lose. The same holds for binary games using $\mathrm{Z}_\triangle$. When $G \geqslant \mathrm{Z}_\triangle$ then MAX can win when MIN has the first move, since MAX prefers G over $\overline{\mathrm{Z}_\triangle} = \mathrm{Z}_\triangle$. When $G \leqslant \mathrm{Z}_\triangle$ then MIN can win when MAX has the first move for the same reason.

Whereas canonical forms can be used for conjunctions and disjunctions, the same is not true for the game $G \equiv H$, because Theorem 7.3.3 does not hold for the equivalence operator '$\equiv$'. It already fails in the base cases with just T and F. We cannot use $(G \vee \overline{H}) \wedge (\overline{G} \vee H)$ instead of $G \equiv H$ because the four components would not be independent; each move would affect two components. The game $G \equiv H$ still has a combinatorial value and a canonical form, simply because

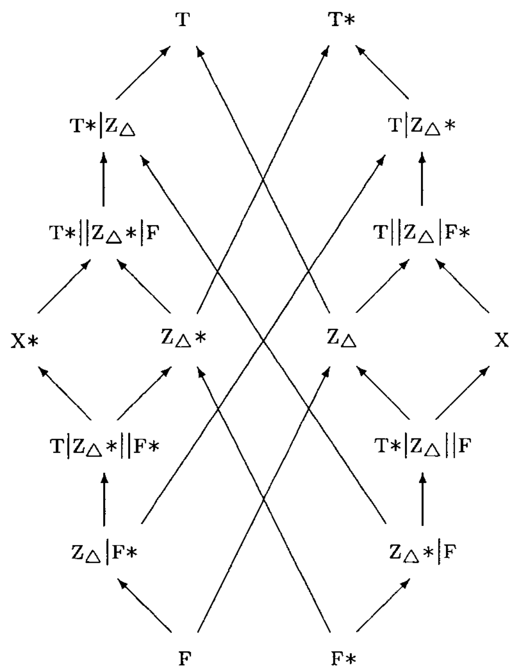
$$G \equiv H = \{G^L \equiv H,\ G \equiv H^L \mid G^R \equiv H,\ G \equiv H^R\}.$$

In particular, it is easy to see by induction that $(G \equiv \mathrm{T}) = G$ and therefore also $(G \equiv \mathrm{T}*) = (G* \equiv \mathrm{T}) = G*$. For division games and GAME-SAT we have that G and $G \equiv \mathrm{F}$ are played on the negation of each other's formula, but this is not the same as playing $\overline{G}$, as they do not interchange Left's and Right's options.

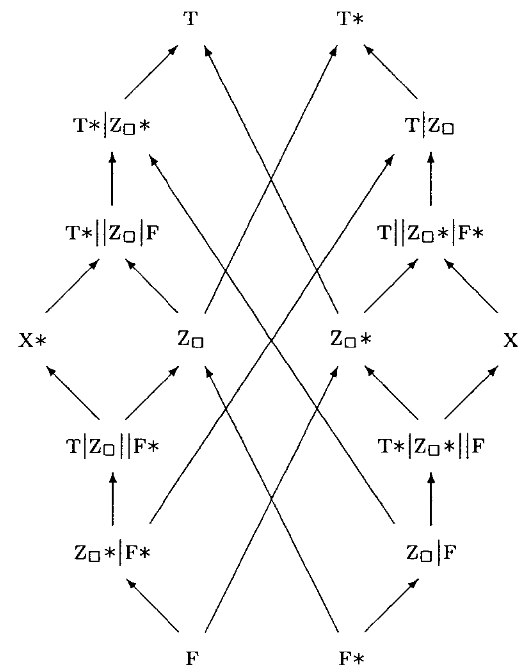
In general the canonical form of $G \equiv H$, unlike the canonical forms of $G \vee H$ and $G \wedge H$, cannot be determined from the canonical forms of G and H alone. An example is formed by the games $\mathrm{F} \equiv \{\mathrm{T}|\mathrm{F}\}$ and $\mathrm{F} \equiv \{\mathrm{T},\mathrm{F}|\mathrm{T},\mathrm{F}\}$, respectively equal to $\{\mathrm{F}|\mathrm{T}\}$ and $\{\mathrm{T}|\mathrm{F}\}$.

Definition 7.6.1. A binary game is **elementary** if it lasts at most one move. It is **decomposable** if it is elementary, or the conjunction or disjunction of games that are themselves decomposable.

With this terminology, the values and canonical forms for decomposable games can be determined from the values of their components.



7.1a: Division Games



7.1b: GAME-SAT

Figure 7.1: Partial order diagram for canonical forms in decomposable games. Arrows point towards greater values. Note: $\mathrm{X} = \{\mathrm{T}|\mathrm{F}\}$, $\mathrm{Z}_\triangle = \{\mathrm{F}|\mathrm{T}\}$, $\mathrm{Z}_\square = \{\mathrm{X}|\mathrm{X}\}$; $\mathrm{X}* = \{\mathrm{T}*|\mathrm{F}*\}$, $\mathrm{Z}_\triangle* = \{\mathrm{Z}_\triangle|\mathrm{Z}_\triangle\}$, $\mathrm{Z}_\square* = \{\mathrm{Z}_\square|\mathrm{Z}_\square\}$.

7.7 Canonical Forms for Division Games and GAME-SAT

Only sixteen different canonical forms occur in decomposable division games. The partial order diagram of these forms is shown in Figure 7.1a. Even games are on the left hand side, odd games are on the right. Adding a star corresponds to reflecting in the vertical axis, and negating a game corresponds to reflecting in the horizontal axis. Table 7.9.1 in Section 7.9 lists some examples of division games with these values.

Negating the formula of a division game does not always correspond to negating the canonical form of the binary game. Playing on the negation of a formula corresponds to playing the game $G \equiv \mathrm{F}$, whose value may not be uniquely determined by G's canonical form. For the examples in Table 7.9.1, negating the formula switches the pairs $\mathrm{X}, \mathrm{Z}_\triangle$ and $\mathrm{X}*, \mathrm{Z}_\triangle*$, and negates all other canonical forms.

For non-decomposable division games it might be possible to construct other values. Testing all combinations of same-parity left and right options, three new candidates present themselves:

- If the game is even and positive, and MAX has an option T$*$, then the value is T. But if MAX does not have an option T$*$ then the value is $+1$, not T.
- If the game is even and negative, and MIN has an option F$*$, then the value is F. But if MIN does not have an option F$*$ then the value is -1, not F.
- If the game is even and zero.

The simplest examples are $\{\text{z}|\text{x}\} = +1$ and $\{\text{x}|\text{z}\} = -1$. These values are $+1$ and -1 regardless of the choice of integer for T, so they are in some sense not *identical* to T. But we are free to choose T $= +1$, in which case the values $+1$ and -1 are not new. The third case would however be new regardless, because it would be an even zero, while Figure 7.1a only contains an odd zero. Computer searches of nondecomposable division games have not yet turned up any instances of an even zero.

These three new candidate values cannot arise from the conjunction or disjunction of two of the values in Figure 7.1, as evidenced by Table 7.9.2 in Section 7.9. They do occur when equivalences are introduced, for instance when X and Z are elementary division games then X $\equiv$ Z is equal to $+1$ and X $\equiv$ X and Z $\equiv$ Z are both equal to -1.

Theorem 7.7.1. Any decomposable division game can be represented by an equivalent game with one of the canonical forms of Figure 7.1a, which can be done using at most three variables.

The fact that no more than three variables are needed can be verified from Table 7.9.1. Statements about decomposable division games can be verified by checking only a small number of cases. For instance:

Theorem 7.7.2. An odd decomposable division game is equal to $\text{z}_\triangle$ if and only if MAX does not have an option T and MIN does not have an option F. For even decomposable division games the same holds with $\text{z}_\triangle *$, T$*$, and F$*$.

Decomposable instances for GAME-SAT only give rise to the canonical forms T, F, X, and their three starred counterparts. In any GAME-SAT position both players have the same options, so we cannot have a canonical form $G = \{G^L, \dots | G^R, \dots\}$ with $G^L < G^R$, as both G^L and G^R would then be dominated. Therefore in particular $\text{z}_\triangle$ does not occur. However, GAME-SAT played on the formula $x \equiv y$ is the game $\{\text{x}|\text{x}\}$ whose combinatorial value is zero. This game shall be denoted $\text{z}_\square$, being the "even zero" as opposed to the "odd zero" $\text{z}_\triangle$.

Definition 7.7.3. A binary game is **semi-decomposable** if it is elementary, or the equivalence of two elementary games, or the conjunction or disjunction of games that are themselves semi-decomposable.

Such equivalences can be allowed in GAME-SAT because they only involve elementary games, so that their canonical values are known to be equal to those of decomposable games. This is different from the situation in division games where the equivalence of two elementary games is $+1$ or -1; recall that any equivalence does have a value, but it is in general not uniquely determined by the values of the components.

The winner of a GAME-SAT instance can be found by comparing the game to $\text{z}_\square$. Figure 7.1b displays the partial order between the sixteen canonical GAME-SAT forms. As can be seen, the figure is identical to Figure 7.1a with $\text{Z}_\triangle$ replaced by $\text{Z}_\square *$. Section 7.9 contains the multiplication tables as well as examples of GAME-SAT instances with given canonical forms. The canonical forms need up to four variables, essentially because specifying a combinatorial zero requires one more variable in GAME-SAT than it does in division games.

Negating the formula of a GAME-SAT instance corresponds to negating the canonical form, because the left and right options are always the same in GAME-SAT, so interchanging them makes no difference. When testing combinations of options, candidate values $+1$ and -1 appear as well, but they seem more implausible as they would occur as odd games whereas T and F are even. The candidate value 0 for the "wrong" parity does not occur, as it would require an undominated left option that is smaller than an undominated right option, which is impossible in GAME-SAT where any left option is also a right option and vice versa.

Theorem 7.7.4. Any semi-decomposable GAME-SAT instance can be represented by an equivalent game with one of the canonical forms of Figure 7.1b, which can be done using at most four variables.

And therefore in particular:

Theorem 7.7.5. Only even GAME-SAT instances can be misère.

The latter theorem would also follow from the stronger statement that the loser can always force the game to be decided only on the last move, as proved for instance for the game of misère Hex by Lagarias and Sleator [59]. This is however *not* true for GAME-SAT; consider for instance the misère game $\langle \mathbb{T}^4, \xi \mapsto (\xi_0 \equiv \xi_1) \wedge (\xi_2 \equiv \xi_3) \rangle$, which MIN, when moving second, can decide on the second move.

Computer searches have not uncovered any other values for general instances in GAME-SAT either, nor indeed for set colouring games with more than two colours.

Theorem 7.7.6. An even semi-decomposable division game is equal to $\text{Z}_\square$ if and only if MAX does not have an option T$*$ and MIN does not have an option F$*$. For odd semi-decomposable division games the same holds with $\text{Z}_\square *$, T, and F.

This too can be confirmed by checking all possible cases.

7.8 Strategies

Let $\mathcal{Q} = \{\langle X^{\mathcal{S}_i}, f_i \rangle\}_{i \in \mathbb{Z}_k}$ and $\Diamond \in \{\square, \triangle\}$. For the conjunctive partition game $\Gamma = \langle \langle \mathcal{Q}, \wedge \rangle \rangle$ Theorems 6.4.1-6.4.4 can be restated as follows:

i. If there is a component Γ_i such that $\Gamma_i * \Diamond \leqslant 0$ then $\Gamma * \Diamond \leqslant 0$. The same holds with $\lhd\!\shortmid$ instead of $\leqslant$.

ii. If for all components we have $\Gamma_i * \Box \geqslant 0$, then $\Gamma * \Box \geqslant 0$.

iii. If there are two different components i, j such that $\Gamma_i * \Diamond \lhd\!\mid 0$ and $\Gamma_j * \lhd\!\mid 0$, then $\Gamma * \Diamond * \leqslant 0$.

iv. If there is one component i such that $\Gamma_i * \triangle \mid\!\rhd 0$, and for all other components j we have $\Gamma_j * \Box \geqslant 0$, then $\Gamma * \triangle \mid\!\rhd 0$.

Each of these observations has an accompanying winning strategy, as described in the discussion in Section 6.4. The corresponding cases for disjunctive partition games are obtained by interchanging MAX and MIN, and interchanging $>$ and $<$.

In general, then, it would be good to know how each component is related to 0, both as an even as well as an odd game. Which combinations are possible? If $\Gamma \geqslant 0$ then $\Gamma * \mid\!\rhd 0$, because MAX can win $\Gamma *$ when going first by taking the star. Similarly, if $\Gamma \leqslant 0$ then $\Gamma * \lhd\!\mid 0$. That leaves nine possible combinations: $\Gamma * \Box$ and $\Gamma * \triangle$ are both positive or both negative, or one of them is fuzzy.

Three of these combinations have already been encountered in division games as well as GAME-SAT, namely the ones where G and $G*$ are both positive, both negative, or both fuzzy. Three combinations have so far only been seen in GAME-SAT, namely the combinations where $G * \triangle$ is fuzzy and $G * \Box$ is not. These have not been observed in division games, as the only fuzzy odd division game seen so far is X. For the remaining three combinations it is the other way around; having fuzzy $G * \Box$ and nonfuzzy $G * \triangle$ they have only been seen in division games, as the only fuzzy even GAME-SAT seen so far is X$*$.

Table 7.8.1 lists what is known about the conjunction of two independent components for the combinations of the relations of the components to 0. The top part of the table contains the relationships that have been found to occur in practice,[3] the bottom part contains the relationships that are not disproved but have not been observed. All of these unobserved relations involve metagames where MAX does not have the last move.

An example of a somewhat surprising case is the division game played on the formula $(\overline{w} \vee x) \wedge (\overline{y} \vee z)$, with MAX to move first. Both components of the conjunction are fuzzy. It would therefore seem that any opening move by MAX would leave the other component as a first-player win for MIN. Yet MAX wins this conjunction, so MAX even wins the component in which MIN moves first. The winning strategy for MAX is necessarily not a partition strategy, since a partition strategy loses after MIN makes the first move in a fuzzy component.

7.9 Tables

Table 7.9.1 contains examples of division games with given canonical forms. Table 7.9.2 contains the "multiplication tables" for $\vee$ and $\wedge$, with the canonical forms occurring in decomposable division games. Only the even values are listed in the tables; all other combinations can be obtained by using theorem 7.5.1.

[3] See Tables 7.9.2 and 7.9.4 in Section 7.9.

	>,>	||,||	<,<	||,>	||,=	||,<	>,||	=,||	<,||
>,>	>,>	||,||	<,<	||,>	||,=	||,<	>,||	=,||	<,||
||,||	||,||	<,<	<,<	||,<	||,<	<,<	<,||	<,||	<,<
>,>	<,<	<,<	<,<	<,<	<,<	<,<	<,<	<,<	<,<
||,>	||,>	||,<	<,<	||,⪅	||,<	⧏,<	||,||	<,||	<,<
||,=	||,=	||,<	<,<	||,<	<,<	<,<	||,<	<,<	<,<
||,<	||,<	<,<	<,<	⧏,<	<,<	<,<	<,<	<,<	<,<
>,||	>,||	<,||	<,<	||,||	||,<	<,<	=,||	=,||	<,||
=,||	=,||	<,||	<,<	<,||	<,<	<,<	=,||	=,||	<,||
<,||	<,||	<,<	<,<	<,<	<,<	<,<	<,||	<,||	<,<

	>,>	||,||	<,<	||,>	||,=	||,<	>,||	=,||	<,||
>,>	=,||	<,	,	<,||	<,<	<,	=,	,	,
||,||	<,	,	,	<,	<,	,	,	,	,
<,<	,	,	,	,	,	,	,	,	,
||,>	<,||	<,	,	<,	<,=	,	<,	,	,
||,=	<,<	<,	,	<,=	||,=	||,	<,	,	,
||,<	<,	,	,	,	||,	,	,	,	,
>,||	=,	,	,	<,	<,	,	,	,	,
=,||	,	,	,	,	,	,	,	,	,
<,||	,	,	,	,	,	,	,	,	,

Table 7.8.1: Observed (top) and unobserved but not disproved (bottom) relations of $G*\square$ and $G*\triangle$ to 0, where G is the conjunction of two independent binary games. Row and column entries list the same for the two components.

canonical form	examples	canonical form	examples
T	TRUE $x \vee y$ $\overline{x} \vee \overline{y}$	$\mathrm{T}* = \{\mathrm{T}\|\mathrm{T}\}$	TRUE$*$ $x \vee y \vee z$ $\overline{x} \vee y \vee z$ $\overline{x} \vee \overline{y} \vee z$ $\overline{x} \vee \overline{y} \vee \overline{z}$
$\mathrm{T}*\|\mathrm{Z}_{\triangle}$	$\overline{x} \vee y$	$\mathrm{T}\|\mathrm{Z}_{\triangle}*$	$(\overline{x} \vee y)*$ $\overline{x} \vee (\overline{y} \wedge z)$
$\mathrm{T}*\|\|\mathrm{Z}_{\triangle}*\|\mathrm{F}$	$(x \vee (\overline{y} \wedge z))*$ $w \vee (\overline{x} \wedge (\overline{y} \vee z))$	$\mathrm{T}\|\|\mathrm{Z}_{\triangle}\|\mathrm{F}*$	$x \vee (\overline{y} \wedge z)$
$\mathrm{X}* = \{\mathrm{T}*\|\mathrm{F}*\}$	$x*$ $w \vee (x \wedge y \wedge z)$ $w \wedge (x \vee y \vee z)$	$\mathrm{X} = \{\mathrm{T}\|\mathrm{F}\}$	x
$\mathrm{Z}_{\triangle}* = \{\mathrm{Z}_{\triangle}\|\mathrm{Z}_{\triangle}\}$	$\overline{x}*$ $(\overline{w} \wedge x) \vee (\overline{y} \wedge z)$ $(\overline{w} \vee x) \wedge (\overline{y} \vee z)$	$\mathrm{Z}_{\triangle} = \{\mathrm{F}\|\mathrm{T}\}$	$\overline{x}$
$\mathrm{T}\|\mathrm{Z}_{\triangle}*\|\|\mathrm{F}*$	$(x \wedge (\overline{y} \vee z))*$ $w \wedge (\overline{x} \vee (\overline{y} \wedge z))$	$\mathrm{T}*\|\mathrm{Z}_{\triangle}\|\|\mathrm{F}$	$x \wedge (\overline{y} \vee z)$
$\mathrm{Z}_{\triangle}\|\mathrm{F}*$	$\overline{x} \wedge y$	$\mathrm{Z}_{\triangle}*\|\mathrm{F}$	$(\overline{x} \wedge y)*$ $\overline{x} \wedge (\overline{y} \vee z)$
F	FALSE $x \wedge y$ $\overline{x} \wedge \overline{y}$	$\mathrm{F}* = \{\mathrm{F}\|\mathrm{F}\}$	FALSE$*$ $x \wedge y \wedge z$ $\overline{x} \wedge y \wedge z$ $\overline{x} \wedge \overline{y} \wedge z$ $\overline{x} \wedge \overline{y} \wedge \overline{z}$

Table 7.9.1: Examples of division games played on various formulas. A '$*$' in a formula indicates a dead variable.

$\vee$	T	$T{*}\mid Z_\triangle$	$T{*}\Vert Z_\triangle{*}\mid F$	$Z_\triangle{*}$	$X{*}$	$T\mid Z_\triangle{*}\Vert F{*}$	$Z_\triangle\mid F{*}$	F
T	T	T	T	T	T	T	T	T
$T{*}\mid Z_\triangle$	T	T	T	T	T	T	T	$T{*}\mid Z_\triangle$
$T{*}\Vert Z_\triangle{*}\mid F$	T	T	T	T	T	T	$T{*}\mid Z_\triangle$	$T{*}\Vert Z_\triangle{*}\mid F$
$Z_\triangle{*}$	T	T	T	T	$T{*}\mid Z_\triangle$	$T{*}\mid Z_\triangle$	$T{*}\mid Z_\triangle$	$Z_\triangle{*}$
$X{*}$	T	T	T	$T{*}\mid Z_\triangle$	T	$T{*}\mid Z_\triangle$	$T{*}\Vert Z_\triangle{*}\mid F$	$X{*}$
$T\mid Z_\triangle{*}\Vert F{*}$	T	T	T	$T{*}\mid Z_\triangle$	$T{*}\mid Z_\triangle$	$T{*}\mid Z_\triangle$	$T{*}\Vert Z_\triangle{*}\mid F$	$T\mid Z_\triangle{*}\Vert F{*}$
$Z_\triangle\mid F{*}$	T	T	$T{*}\mid Z_\triangle$	$T{*}\mid Z_\triangle$	$T{*}\Vert Z_\triangle{*}\mid F$	$T{*}\Vert Z_\triangle{*}\mid F$	$Z_\triangle{*}$	$Z_\triangle\mid F{*}$
F	T	$T{*}\mid Z_\triangle$	$T{*}\Vert Z_\triangle{*}\mid F$	$Z_\triangle{*}$	$X{*}$	$T\mid Z_\triangle{*}\Vert F{*}$	$Z_\triangle\mid F{*}$	F

$\wedge$	T	$T{*}\mid Z_\triangle$	$T{*}\Vert Z_\triangle{*}\mid F$	$X{*}$	$Z_\triangle{*}$	$T\mid Z_\triangle{*}\Vert F{*}$	$Z_\triangle\mid F{*}$	F
T	T	$T{*}\mid Z_\triangle$	$T{*}\Vert Z_\triangle{*}\mid F$	$X{*}$	$Z_\triangle{*}$	$T\mid Z_\triangle{*}\Vert F{*}$	$Z_\triangle\mid F{*}$	F
$T{*}\mid Z_\triangle$	$T{*}\mid Z_\triangle$	$Z_\triangle{*}$	$T\mid Z_\triangle{*}\Vert F{*}$	$T\mid Z_\triangle{*}\Vert F{*}$	$Z_\triangle\mid F{*}$	$Z_\triangle\mid F{*}$	F	F
$T{*}\Vert Z_\triangle{*}\mid F$	$T{*}\Vert Z_\triangle{*}\mid F$	$T\mid Z_\triangle{*}\Vert F{*}$	$Z_\triangle\mid F{*}$	$Z_\triangle\mid F{*}$	$Z_\triangle\mid F{*}$	F	F	F
$X{*}$	$X{*}$	$T\mid Z_\triangle{*}\Vert F{*}$	$Z_\triangle\mid F{*}$	F	$Z_\triangle\mid F{*}$	F	F	F
$Z_\triangle{*}$	$Z_\triangle{*}$	$Z_\triangle\mid F{*}$	$Z_\triangle\mid F{*}$	$Z_\triangle\mid F{*}$	F	F	F	F
$T\mid Z_\triangle{*}\Vert F{*}$	$T\mid Z_\triangle{*}\Vert F{*}$	$Z_\triangle\mid F{*}$	F	F	F	F	F	F
$Z_\triangle\mid F{*}$	$Z_\triangle\mid F{*}$	F	F	F	F	F	F	F
F	F	F	F	F	F	F	F	F

Table 7.9.2: Multiplication table for $\vee$ (top) and $\wedge$ (bottom). Note that the order of $X{*}$ and $Z_\triangle{*}$ is reversed between the tables, for cosmetic purposes.

Tables 7.9.3 and 7.9.4 list the same for decomposable GAME-SAT. As can be seen in Table 7.9.4, the multiplication tables for the GAME-SAT positions look slightly different from those for the division game positions.

To conclude, Table 7.9.5 lists observed frequencies for set colouring games of various dimensions with two and three colours. Where integers are listed, the statistics comprise an enumeration of all possible games of the given type. Percentage statistics were obtained by random sampling of the space of games.

In CGT, a game in which both players always have the same moves available is called an **impartial game**. It has been shown that every impartial game is essentially a variant of the game of Nim [13]. The CGT analysis of set colouring games makes them *almost* impartial: both players have the same options available in all positions except final positions.

<table>
<tr><th>canonical form</th><th>examples</th><th>canonical form</th><th>examples</th></tr>
<tr><td>T</td><td>TRUE
$x \vee y$</td><td>T* = {T|T}</td><td>TRUE*
$x \vee y \vee z$</td></tr>
<tr><td>T*|Z□*</td><td>$(x \vee (y \equiv z))*$</td><td>T|Z□</td><td>$x \vee (y \equiv z)$</td></tr>
<tr><td>T*||Z□|F</td><td>$w \vee (x \wedge (y \equiv z))$</td><td>T||Z□*|F*</td><td>$(w \vee (x \wedge (y \equiv z)))*$</td></tr>
<tr><td>X* = {T*|F*}</td><td>$x*$
$w \wedge (x \vee y \vee z)$
$w \vee (x \wedge y \wedge z)$</td><td>X = {T|F}</td><td>x
$x \vee (y \wedge z)$
$x \wedge (y \vee z)$</td></tr>
<tr><td>Z□ = {X|X}</td><td>$x \equiv y$</td><td>Z□* = {Z□|Z□}</td><td>$(x \equiv y)*$
$x \equiv y \equiv z$</td></tr>
<tr><td>T|Z□||F*</td><td>$w \wedge (x \vee (y \equiv z))$</td><td>T*|Z□*||F</td><td>$(w \wedge (x \vee (y \equiv z)))*$</td></tr>
<tr><td>Z□*|F*</td><td>$(x \wedge (y \equiv z))*$</td><td>Z□|F</td><td>$x \wedge (y \equiv z)$</td></tr>
<tr><td>F</td><td>FALSE
$x \wedge y$</td><td>F* = {F|F}</td><td>FALSE*
$x \wedge y \wedge z$</td></tr>
</table>

Table 7.9.3: Examples of GAME-SAT played on decomposable formulas.

<table>
<tr><th>∨</th><th>T</th><th>T*|Z□*</th><th>T*||Z□|F</th><th>X*</th><th>Z□</th><th>T|Z□||F*</th><th>Z□*|F*</th><th>F</th></tr>
<tr><td>T</td><td>T</td><td>T</td><td>T</td><td>T</td><td>T</td><td>T</td><td>T</td><td>T</td></tr>
<tr><td>T*|Z□*</td><td>T</td><td>T</td><td>T</td><td>T</td><td>T*|Z□*</td><td>T*|Z□*</td><td>T*|Z□*</td><td>T*|Z□*</td></tr>
<tr><td>T*||Z□|F</td><td>T</td><td>T</td><td>T</td><td>T</td><td>T*|Z□*</td><td>T*|Z□*</td><td>T*|Z□*</td><td>T*||Z□|F</td></tr>
<tr><td>X*</td><td>T</td><td>T</td><td>T</td><td>T</td><td>T*|Z□*</td><td>T*|Z□*</td><td>T*||Z□|F</td><td>X*</td></tr>
<tr><td>Z□</td><td>T</td><td>T*|Z□*</td><td>T*|Z□*</td><td>T*|Z□*</td><td>Z□</td><td>Z□</td><td>Z□</td><td>Z□</td></tr>
<tr><td>T|Z□||F*</td><td>T</td><td>T*|Z□*</td><td>T*|Z□*</td><td>T*|Z□*</td><td>Z□</td><td>Z□</td><td>Z□</td><td>T|Z□||F*</td></tr>
<tr><td>Z□*|F*</td><td>T</td><td>T*|Z□*</td><td>T*|Z□*</td><td>T*||Z□|F</td><td>Z□</td><td>Z□</td><td>Z□</td><td>Z□*|F*</td></tr>
<tr><td>F</td><td>T</td><td>T*|Z□*</td><td>T*||Z□|F</td><td>X*</td><td>Z□</td><td>T|Z□||F*</td><td>Z□*|F*</td><td>F</td></tr>
</table>

<table>
<tr><th>∧</th><th>T</th><th>T*|Z□*</th><th>T*||Z□|F</th><th>Z□</th><th>X*</th><th>T|Z□||F*</th><th>Z□*|F*</th><th>F</th></tr>
<tr><td>T</td><td>T</td><td>T*|Z□*</td><td>T*||Z□|F</td><td>Z□</td><td>X*</td><td>T|Z□||F*</td><td>Z□*|F*</td><td>F</td></tr>
<tr><td>T*|Z□*</td><td>T*|Z□*</td><td>Z□</td><td>Z□</td><td>Z□</td><td>T|Z□||F*</td><td>Z□*|F*</td><td>Z□*|F*</td><td>F</td></tr>
<tr><td>T*||Z□|F</td><td>T*||Z□|F</td><td>Z□</td><td>Z□</td><td>Z□</td><td>Z□*|F*</td><td>Z□*|F*</td><td>Z□*|F*</td><td>F</td></tr>
<tr><td>Z□</td><td>Z□</td><td>Z□</td><td>Z□</td><td>Z□</td><td>Z□*|F*</td><td>Z□*|F*</td><td>Z□*|F*</td><td>F</td></tr>
<tr><td>X*</td><td>X*</td><td>T|Z□||F*</td><td>Z□*|F*</td><td>Z□*|F*</td><td>F</td><td>F</td><td>F</td><td>F</td></tr>
<tr><td>T|Z□||F*</td><td>T|Z□||F*</td><td>Z□*|F*</td><td>Z□*|F*</td><td>Z□*|F*</td><td>F</td><td>F</td><td>F</td><td>F</td></tr>
<tr><td>Z□*|F*</td><td>Z□*|F*</td><td>Z□*|F*</td><td>Z□*|F*</td><td>Z□*|F*</td><td>F</td><td>F</td><td>F</td><td>F</td></tr>
<tr><td>F</td><td>F</td><td>F</td><td>F</td><td>F</td><td>F</td><td>F</td><td>F</td><td>F</td></tr>
</table>

Table 7.9.4: Multiplication table for ∨ (top) and ∧ (bottom), with Z□ = {X|X}. Note that the order of X* and Z□ is reversed between the two tables.

		game dimension 0	1	2	3	4	5	6
		$\lvert\check{X}\rvert = 2$						
F	T	1	-	5	-	9849	-	3.4%
X*		-	-	4	-	5032	-	0.4%
Z□		-	-	2	-	20006	-	75.3%
Z□*|F*	T*|Z□*	-	-	-	-	2304	-	3.2%
T|Z□||F*	T*||Z□|F	-	-	-	-	8096	-	5.6%
F*	T*	-	1	-	25	-	1.4%	-
X		-	2	-	110	-	20.8%	-
Z□*		-	-	-	8	-	11.4%	-
Z□|F	T|Z□	-	-	-	44	-	28.8%	-
T*|Z□*||F	T||Z□*|F*	-	-	-	-	-	3.7%	-
		$\lvert\check{X}\rvert = 3$						
F	T	1	-	163	-	8.9%	-	
X*		-	-	84	-	0.7%	-	
Z□		-	-	102	-	72.6%	-	
Z□*|F*	T*|Z□*	-	-	-	-	0.1%	-	
T|Z□||F*	T*||Z□|F	-	-	-	-	4.4%	-	
F*	T*	-	1	-	1.4%	-		-
X		-	6	-	73.1%	-		-
Z□*		-	-	-	0.2%	-		-
Z□|F	T|Z□	-	-	-	12.0%	-		-
T*|Z□*||F	T||Z□*|F*	-	-	-	-	-		-

Table 7.9.5: Frequencies of values for set colouring games.

Chapter 8

Superrational Play

Section 5.2 introduced optimal colourings of elements, being the "best possible colouring" of a given element that a player can achieve, with the associated notion of a rational move. This can be generalized to sets of elements that have a "best possible colouring", leading to "superrational play". In many cases this allows simplification of the analysis of a game.

8.1 Optimal Colourings

An optimal colouring occurs when some subset of the elements of a set colouring game is coloured in the best possible way for one of the players. Any game $\Gamma = \langle X^{\mathcal{S}}, f \rangle$ can be said to impose a partial order on all colourings of $\mathcal{S}$ as follows:

Definition 8.1.1. Let $\Gamma = \langle X^{\mathcal{S}}, f \rangle$ and $\psi_0, \psi_1 \in X^{\mathcal{S}}$. Then $\psi_0 \geqslant_\Gamma \psi_1$ whenever $\Gamma/\psi_0 \geqslant \Gamma/\psi_1$.

Observation i: Equivalently, $\psi_0 \geqslant_\Gamma \psi_1$ if and only if $f(\psi^*\overline{\psi_0}) \geqslant f(\psi^*\overline{\psi_1})$ for all $\psi^* \in \check{X}^{\mathcal{S}}$.

Observation ii: If $\psi_0 \geqslant_\Gamma \psi_1$ and $\psi_1 \succcurlyeq \psi_2$ then $\psi_0 \geqslant_\Gamma \psi_2$.

The preferable moves of Definition 5.2.1 are a special case of this partial ordering. If $\psi_0 \geqslant_\Gamma \psi_1$ then, if no parity issues arise, MAX prefers ψ_0 over ψ_1, even when $\mathcal{A}(\psi_0) \neq \mathcal{A}(\psi_1)$. This naturally leads to the notion of optimal colourings. The following definition is the generalization of rational moves:

Definition 8.1.2. Let $\Gamma = \langle X^{\mathcal{S}}, f \rangle$ and $\psi \in X^{\mathcal{S}}$. Then ψ is **a maximal colouring** if $\psi \geqslant_\Gamma \psi'$ for all $\psi' \in X^{\mathcal{S}}$ with $\mathcal{A}(\psi') = \mathcal{A}(\psi)$, and ψ is **a minimal colouring** if $\psi \leqslant_\Gamma \psi'$ for all such ψ'. If ψ is a maximal colouring or a minimal colouring then ψ is **an optimal colouring**.

Observation i: Colouring ψ is maximal if and only if $\Gamma/\psi \geqslant \Gamma$, and ψ is minimal if and only if $\Gamma/\psi \leqslant \Gamma$.

Observation ii: Let $\psi, \psi' \in X^{\mathcal{S}}$ with $\mathcal{A}(\psi') \subseteq \mathcal{A}(\psi)$. If ψ is a maximal colouring then $\Gamma/\psi \geqslant \Gamma/\psi'$. If ψ is a minimal colouring then $\Gamma/\psi \leqslant \Gamma/\psi'$.

For any given set $\mathcal{S}' \subseteq \mathcal{S}$ it is possible that there is more than one maximal colouring, and if ψ and ψ' are both maximal colourings with $\mathcal{A}(\psi) = \mathcal{A}(\psi')$ then of course $\Gamma/\psi = \Gamma/\psi'$. It is possible that $\mathcal{S}'$ has a maximal colouring but no minimal colouring, or vice versa, or neither. For instance, in $\langle \mathbb{T}^2, \xi \mapsto (\xi_0 = \chi) \vee (\xi_0 = \xi_1)\rangle$ the set $\mathcal{S}' = \{0\}$ has a maximal colouring, namely $\xi_0 = \chi$, but no minimal colouring. In $\langle \mathbb{T}^2, \xi \mapsto \xi_0 = \xi_1\rangle$ no subset of size 1 has an optimal colouring. Every pure colouring of the entire set is optimal.

If a certain subset $\mathcal{S}'$ consists of dead elements then every pure colouring of $\mathcal{S}'$ is both maximal and minimal. If Γ is isotone then $\mathrm{T}^{\mathcal{S}'}$ is a maximal colouring and $\mathrm{F}^{\mathcal{S}'}$ is a minimal colouring for any subset $\mathcal{S}'$. This follows directly from Theorem 5.5.3. It is trivial to see that combining two like optimal colourings produces another optimal colouring of the same kind:

Theorem 8.1.3. Let $\Gamma = \langle X^{\mathcal{S}}, f\rangle$ and $\psi_0, \psi_1 \in X^{\mathcal{S}}$. If ψ_0 and ψ_1 are both maximal colourings then $\psi_0\psi_1$ is also a maximal colouring. If ψ_0 and ψ_1 are both minimal colourings then $\psi_0\psi_1$ is also a minimal colouring.

Any optimal colouring may contain nonmonotone elements, but should assign the appropriate optimal colourings to any monotone elements that appear in it. An example of a maximal colouring containing nonmonotone elements is the colouring $\psi = (+,+) \in \mathbb{B}^2$ in the function $\xi \mapsto (\xi_0 = \xi_1) \wedge f(\xi)$ where f is some function in which the elements 0 and 1 are dead. Neither element 0 nor element 1 has an optimal colouring itself; this also shows that removing an element from an optimal colouring may destroy its optimality, so the property of being an optimal colouring is not hereditary.

8.2 Metagames Based on Optimal Colourings

Based on a partial colouring ψ, two particular games can be defined. These games each involve one of the players attempting to achieve a colouring of $\mathcal{A}(\psi)$ that is at least as advantageous as ψ itself.

Definition 8.2.1. Let $\Gamma = \langle X^{\mathcal{S}}, f\rangle$ and $\psi \in X^{\mathcal{S}}$. Define the game $\Gamma_\psi^+ \stackrel{\text{def}}{=\!=} \langle X^{\mathcal{A}(\psi)}, f_\psi^+\rangle$ with $f_\psi^+ : \check{X}^{\mathcal{A}(\mathcal{S})} \to \mathbb{B}$ defined as

$$\xi \mapsto \begin{cases} +1 & \text{if } \Gamma/\xi \geqslant \Gamma/\psi, \\ -1 & \text{otherwise.} \end{cases}$$

Analogously define $\Gamma_\psi^- \stackrel{\text{def}}{=\!=} \langle X^{\mathcal{A}(\psi)}, f_\psi^-\rangle$ with $f_\psi^- : \check{X}^{\mathcal{A}(\psi)} \to \mathbb{B}$ defined as

$$\xi \mapsto \begin{cases} -1 & \text{if } \Gamma/\xi \leqslant \Gamma/\psi, \\ +1 & \text{otherwise.} \end{cases}$$

Observation i: In Γ^+_ψ we have $f^+_\psi(\psi') = -1 \iff \exists_{\psi^* \in \check{X}^S}[f(\psi^*\overline{\psi}) > f(\psi^*\psi')]$. Similarly for Γ^-_ψ we have $f^-_\psi(\psi') = +1 \iff \exists_{\psi^* \in \check{X}^S}[f(\psi^*\overline{\psi}) < f(\psi^*\psi')]$.

Observation ii: By Observation 6.1.1:i, if $\psi \in \check{X}^S$ then $f(\psi) = +1 \iff \Gamma^+_\psi = \Gamma$ and $f(\psi) = -1 \iff \Gamma^-_\psi = \Gamma$.

The specifications $\Gamma/\xi \geqslant \Gamma/\psi$ and $\Gamma/\xi \leqslant \Gamma/\psi$ are the natural generalizations of the notion of preferable colourings as introduced in Section 5.2. The generalization of a rational move is a game Γ^+_ψ or Γ^-_ψ where ψ is optimal.

Definition 8.2.2. Let $\Gamma = \langle X^S, f\rangle$ and $S' \subseteq S$. The game $\Gamma^+_{S'}$ is defined as Γ^+_ψ where $\psi \in \check{X}^S$ is some maximal colouring. The game $\Gamma^-_{S'}$ is defined as Γ^-_ψ where $\psi \in \check{X}^S$ is some minimal colouring.

Observation i: The game $\Gamma^+_{S'}$ is played on S' with the outcome function

$$\xi \mapsto \begin{cases} +1 & \text{if } \Gamma/\psi \geqslant \Gamma, \\ -1 & \text{otherwise} \end{cases}$$

which is equivalent to

$$\xi \mapsto \begin{cases} +1 & \text{if} \forall_{\psi^* \in \check{X}^S}[f(\psi * \xi) \geqslant f(\psi^*)], \\ -1 & \text{if} \exists_{\psi^* \in \check{X}^S}[f(\psi^*\xi) < f(\psi^*)]. \end{cases}$$

Similarly, $\Gamma^-_{S'}$ is played with the outcome function that returns $+1$ if and only if $\Gamma/\xi \leqslant \Gamma$, which is equivalent to $f(\psi^*\xi) \leqslant f(\psi^*)$ for all $\psi^* \in \check{X}^S$.

When one player can indeed achieve a colouring of $\mathcal{A}(\psi)$ that is preferable to ψ, and in addition achieve a win starting from ψ, this is sufficient to win the overall game.

Theorem 8.2.3. Let $\Gamma = \langle X^S, f\rangle$ and $\psi \in X^S$. Then

$$\Gamma^+_\psi \wedge \Gamma/\psi \leqslant \Gamma,$$

$$\Gamma^-_\psi \vee \Gamma/\psi \geqslant \Gamma.$$

If ψ is a maximal colouring, then with Observation 8.1.2:i we obtain $\Gamma^+_\psi \wedge \Gamma/\psi \leqslant \Gamma \leqslant \Gamma/\psi$. Intuitively this hints that if MAX can win Γ^+_ψ then $\Gamma/\psi = \Gamma$. For minimal colourings we have the analogous expression $\Gamma^-_\psi \vee \Gamma/\psi \geqslant \Gamma \geqslant \Gamma/\psi$, so if MIN can win Γ^-_ψ then $\Gamma/\psi = \Gamma$. The following sections will explore these properties more precisely.

8.3 Substitutions

The main theme of this chapter will be the simplification of games by "filling in" a certain colouring; that is, colouring some of the elements and carrying on the game from there.

Definition 8.3.1. Let $\Gamma = \langle X^{\mathcal{S}}, f\rangle$ and $\psi \in X^{\mathcal{S}}$. Replacing Γ with Γ/ψ if $|\mathcal{A}(\psi)|$ is even, or with $(\Gamma/\psi)*$ if $|\mathcal{A}(\psi)|$ is odd, is called **a substitution**.

> *Observation i:* Substituting ψ is equivalent to replacing Γ with $\Gamma/\psi * \mathcal{S}$.

This way, substitutions always preserve the parity of the game by adding a star if necessary. The reason for defining substitutions this way is that substitutions by optimal colourings cannot hurt one of the players.

Theorem 8.3.2. Let $\Gamma = \langle X^{\mathcal{S}}, f\rangle$, let $\psi \in X^{\mathcal{S}}$, and let $\mathfrak{c} \in \mathfrak{C}$. If ψ is a maximal colouring then $\text{MNX}(\Gamma/\psi * \mathcal{S}; \mathfrak{c}) \geqslant \text{MNX}(\Gamma; \mathfrak{c})$. If ψ is a minimal colouring then $\text{MNX}(\Gamma/\psi * \mathcal{S}; \mathfrak{c}) \leqslant \text{MNX}(\Gamma; \mathfrak{c})$.

> *Corollary i:* Substituting a maximal colouring cannot hurt MAX, and substituting a minimal colouring cannot hurt MIN.

From this we arrive at the notion of a capture, which is a case where a particular substitution cannot hurt either player. Obviously, if this occurs, then the substitution can be made without altering the minimax value of the game.

8.4 Capture

In Hex situations often occur where a set of cells, despite actually being empty, are already "virtually conquered" by one of the players. This happens when one of the players can always play in such a way as to reach a beneficial optimal colouring of those cells. An example is Hayward's edge triangle as shown in Figure 1.4 of Section 1.3. This situation has been defined in Hex as *capturing* a set [49, 48]. The definition will be extended to set colouring games here.

Consider some game $\Gamma = \langle X^{\mathcal{S}}, f\rangle$ where $|\mathcal{S}|$ is even, and some maximal colouring $\psi \in X^{\mathcal{S}}$ where $|\mathcal{A}(\psi)|$ is also even. Player MIN is to move. Theorem 8.2.3 says $\Gamma_\psi^+ \wedge \Gamma/\psi \leqslant \Gamma$, so that Theorem 6.4.2 ensures that it would be sufficient for MAX to have second-player wins in both $\Gamma_\psi^+ * \mathcal{S}$ and $\Gamma/\psi * \mathcal{S}$. The games Γ_ψ^+ and Γ/ψ are even, as is $\mathcal{S}$, so this simply means that it is sufficient for MAX to have second-player wins in Γ_ψ^+ and in Γ/ψ. By Observation 8.1.2:i we also have $\Gamma/\psi \geqslant \Gamma$, and since Γ/ψ and Γ have the same parity, Theorem 8.3.2 says that MAX has a second-player win in Γ/ψ if MAX has a second-player win in Γ.

Suppose that MAX does indeed have a second-player win in Γ_ψ^+. In that case, having a second-player win in Γ/ψ would evidently be both sufficient, because $\Gamma_\psi^+ \wedge \Gamma/\psi \leqslant \Gamma$, and necessary, because $\Gamma/\psi \geqslant \Gamma$. Evidently the minimax values of (Γ, MIN) and $(\Gamma/\psi, \text{MIN})$ must be equal. This means that ψ can be substituted without changing the minimax value of the game. It gives MAX a partition strategy to win Γ by winning both Γ_ψ^+ and Γ/ψ independently.

What if $\mathcal{A}(\psi)$ is odd? Then both Γ_ψ^+ and Γ/ψ are odd. However, the substitution property can still be made to work by adding a star to each component. The observation then becomes: if MAX has a second-player win in $\Gamma^+_\psi *$, then the minimax values of $\Gamma/\psi *$ and $\Gamma * *$ are equal. This gives MAX a partition strategy to win $\Gamma * *$, by winning both $\Gamma^+_\psi *$ and $\Gamma/\psi *$ independently. According to Theorem 5.4.2, MAX must then also have a second-player win strategy for Γ. To this end, MAX can *not* use a partition strategy based on the strategies for $\Gamma^+_\psi *$ and $\Gamma/\psi *$, since the two starred moves are in separate components so a partition strategy for $\Gamma * *$ will never answer a star move with another star move. The surprising observation is that the second-player win strategies for the components implies the existence of a second-player win for the overall game, yet the strategy itself is unrelated.

These observations hold when MAX has a second-player win strategy in Γ^+_ψ, if Γ^+_ψ is even, or in $\Gamma^+_\psi *$, if Γ^+_ψ is odd. If Γ itself is odd, then the substitution observation can also be made to work by adding stars. In that case only *one* star is added, since there is only one odd component, which means that the observation then refers to the game $\Gamma *$ and not to Γ itself. In general, we have then that whenever MAX has a second-player win in $\Gamma^+_\psi * \Box$, and MAX goes second, then ψ can be substituted in $\Gamma * \Box$.

Now suppose that Γ^+_ψ is even and Γ/ψ is odd, so that Γ is odd also, and suppose that MAX now moves first. If MAX has a first-player win in Γ/ψ and a second-player win in Γ^+_ψ, then MAX can play a winning move in the Γ/ψ component. This would then leave two even components, both with second-player wins for MAX, and the substitution of the maximal colouring ψ can be made safely. Again by adding stars this observation can be made to work for other combinations of parities as well, so that the general statement is: whenever MAX has a second-player win in $\Gamma^+_\psi * \Box$, and MAX goes first, then ψ can be substituted in $\Gamma * \triangle$.

These properties all rely on MAX having a second-player win in $\Gamma^+_\psi * \Box$. This is the concept of a *capture*.

Definition 8.4.1. Let $\Gamma = \langle X^{\mathcal{S}}, f\rangle$ and $\mathcal{S}' \subseteq \mathcal{S}$. Then $\mathcal{S}'$ is **captured** by MAX if $\text{MNX}(\Gamma^+_\psi * \Box; \text{MIN}) = +1$ for some maximal colouring ψ of $\mathcal{S}'$. Similarly, $\mathcal{S}'$ is captured by MIN if $\text{MNX}(\Gamma^-_\psi * \Box; \text{MAX}) = -1$ for some minimal colouring ψ of $\mathcal{S}'$.

The use of a captured set is that its optimal colouring can be substituted without affecting the minimax value. This simplifies the analysis of the game.

Theorem 8.4.2. Let $\Gamma = \langle X^{\mathcal{S}}, f\rangle$ and $\mathcal{S}' \subseteq \mathcal{S}$, where $\mathcal{S}'$ is captured by player $\mathfrak{c} \in \mathfrak{C}$ with associated optimal colouring $\psi \in \check{X}^{\mathcal{S}'}$. Then $\text{MNX}(\Gamma * \Box; \bar{\mathfrak{c}}) = \text{MNX}(\Gamma/\psi * \Box; \bar{\mathfrak{c}})$, and $\text{MNX}(\Gamma * \triangle; \mathfrak{c}) = \text{MNX}(\Gamma/\psi * \triangle; \mathfrak{c})$.

In other words, if a set of elements is captured with an optimal colouring ψ, then ψ can be substituted when the captor is to move and the game is odd, or when the captor's opponent is to move and the game is even. In the cases with opposite parity, the theorem refers to a substitution in $\Gamma *$, which may not be of any use in analyzing Γ itself. However, if Γ is isotone then the parity does not matter, and the substitution is always safe regardless of whose move it is.

8.5 Domination

In addition to allowing substitutions, captured sets have another strategic consequence. When some move $\mathfrak{m} = \chi^v$ has the side effect of capturing a set $\mathcal{S}'$ that contains v, then $\mathfrak{m}$ is equivalent to playing the best possible moves in all the elements of $\mathcal{S}' + v$ at once. Then $\mathfrak{m}$ must be preferable to *any* move in $\mathcal{S}'$. In Hex this phenomenon was defined as a *dominating move* [49, 48].

The base example to consider in this case is when Γ^+_ψ is odd and Γ/ψ is even, making Γ itself odd. Suppose that MAX has a *first*-player win in Γ^+_ψ, and $\mathfrak{m}$ is a winning first move in Γ^+_ψ. After MAX plays this move, both components have become even, and ψ can be substituted. This substitution must be at least as good for MAX as any other move in $\mathcal{A}(\psi)$, since ψ is a maximal colouring and has the same parity as a single move in $\mathcal{A}(\psi)$. This justification is in fact Observation 8.1.2:ii.

Again by adding stars in order to arrive at these parities, the general concepts emerge as follows.

Definition 8.5.1. Let $\Gamma = \langle X^{\mathcal{S}}, f\rangle$ and $\mathcal{S}' \subseteq \mathcal{S}$. Then $\mathcal{S}'$ is **dominated** by MAX if $\text{MNX}(\Gamma^+_\psi * \triangle; \text{MAX}) = +1$ for some maximal colouring ψ of $\mathcal{S}'$. In that case, any winning move in $(\Gamma^+_\psi, \text{MAX})$ is **a dominating** move for MAX in $\mathcal{S}'$. Similarly, $\mathcal{S}'$ is dominated by MIN if $\text{MNX}(\Gamma^-_\psi * \triangle; \text{MIN}) = -1$ for some minimal colouring ψ of $\mathcal{S}'$, and then any winning move in $(\Gamma^-_\psi * \triangle, \text{MIN})$ is a dominating move for MIN in $\mathcal{S}'$.

The following theorem holds that a dominating move is indeed at least as good as all the moves it dominates.

Theorem 8.5.2. Let $\Gamma = \langle X^{\mathcal{S}}, f\rangle$, and let $\mathcal{S}'$ be dominated by $\mathfrak{c} \in \mathfrak{C}$ with associated optimal colouring $\psi \in \check{X}^{\mathcal{S}'}$ and dominating move $\mathfrak{m}$. Let $\mathfrak{m}' \in \mathfrak{M}(\mathcal{S}')$. Then $\text{NGX}((\Gamma * \triangle)/\mathfrak{m}; \bar{\mathfrak{c}}) \leqslant \text{NGX}((\Gamma * \triangle)/\mathfrak{m}'; \bar{\mathfrak{c}})$.

Whenever a dominated set is found, then only *one* dominating move from that set needs to be considered. If it does not win, then neither will any other move from that set.

As with captured sets, dominating moves give useful information if the parity of the game is right. In fact, in both cases, the useful information in non-isotone games relates to capture or domination by the player who will make the last move of the game. The underlying reason is the same as the "off-parity metagame" of Section 6.4.

8.6 Detecting Optimal Colourings

Identifying captured sets and dominating moves relies on being able to construct games of the type $\Gamma^+_{\mathcal{S}'}$ or $\Gamma^-_{\mathcal{S}'}$. Fortunately there turns out to be an easy way to do this, if the outcome function is given as a CNF or DNF formula. All that needs to be done is to delete all other elements from the formula.

Suppose some function $f : \check{X}^{\mathcal{S}} \to \mathbb{B}$ is given in CNF form as $\mathcal{C}_0 \wedge \mathcal{C}_1 \wedge \ldots \wedge \mathcal{C}_{n-1}$ where each clause $\mathcal{C}_i$ consists of a disjunction of simple equations of the form $\psi_v = \chi$, and suppose that $\mathcal{S}$ is partitioned into $\mathcal{S}'$ and $\mathcal{S}''$. Then f can be rewritten as

$$(\mathcal{C}_0' \vee \mathcal{C}_0'') \wedge (\mathcal{C}_1' \vee \mathcal{C}_1'') \wedge \ldots \wedge (\mathcal{C}_{n-1}' \vee \mathcal{C}_{n-1}'').$$

In this rewrite, $\mathcal{C}_i'$ and $\mathcal{C}_i''$ contain the elements from $\mathcal{S}'$ and $\mathcal{S}''$, respectively, that appear in $\mathcal{C}_i$. When no element from a partition set appears in $\mathcal{C}_i$, then the corresponding subclause $\mathcal{C}_i'$ or $\mathcal{C}_i''$ is set to FALSE. Without loss of generality say that $\mathcal{C}_0', \ldots, \mathcal{C}_{k-1}'$ contain the elements from $\mathcal{S}'$, and $\mathcal{C}_k', \ldots, \mathcal{C}_{n-1}'$ do not contain any elements from $\mathcal{S}'$ and are therefore set to FALSE. Then form the new formula $f' = \mathcal{C}_0' \wedge \mathcal{C}_1' \wedge \ldots \wedge \mathcal{C}_{k-1}'$. This formula is simply obtained from the original formula by removing all elements from $\mathcal{S}''$.

Definition 8.6.1. Let $f : \check{X}^{\mathcal{S}} \to \mathbb{B}$ and $\mathcal{S}' \subseteq \mathcal{S}$. If f is given as a CNF or DNF formula, then the **$\mathcal{S}'$-punctuated** formula is obtained by removing all elements not occurring in $\mathcal{S}'$ from the formula, and subsequently removing all clauses that are empty.

Example 8.6.2. When

$$f(\psi) = (\psi_0 \vee -\psi_1) \wedge (-\psi_0 \vee \psi_2) \wedge (-\psi_2 \vee -\psi_3)$$

we obtain the following punctuated formulas depending on $\mathcal{S}'$:

$$\begin{array}{ll} \mathcal{S}' = \{0\} & f'(\psi) = (\psi_0) \wedge (-\psi_0) \wedge (\varnothing) = \psi_0 \wedge -\psi_0 = \text{FALSE} \\ \mathcal{S}' = \{0,1\} & f'(\psi) = (\psi_0 \vee -\psi_1) \wedge (-\psi_0) \wedge (\varnothing) = (\psi_0 \vee -\psi_1) \wedge -\psi_0 = -\psi_0 \wedge -\psi_1 \\ \mathcal{S}' = \{0,2\} & f'(\psi) = (\psi_0) \wedge (\psi_2) \wedge (-\psi_2) = \text{FALSE} \\ \mathcal{S}' = \{0,1,2\} & f'(\psi) = (\psi_0 \vee -\psi_1) \wedge (-\psi_0 \vee -\psi_2) \wedge (-\psi_2) = -\psi_1 \wedge -\psi_0 \wedge -\psi_2 \end{array}$$

The claim is that any colouring $\psi' \in \check{X}^{\mathcal{S}'}$ that satisfies f' is a maximal colouring under f. This can be appreciated intuitively, for any re-colouring $\psi^*\psi'$ of some $\psi^* \in \check{X}^{\mathcal{S}}$ leaves each $\mathcal{C}_i''$ undisturbed, and sets all nonempty $\mathcal{C}_i'$ to TRUE. In Example 8.6.2 there is no maximal colouring for element 0, but there is a maximal colouring for elements $\{0, 1\}$ together.

Theorem 8.6.3. Let $f : \check{X}^{\mathcal{S}} \to \mathbb{B}$, $\mathcal{S}' \subseteq \mathcal{S}$, and $\psi' \in \check{X}^{\mathcal{S}'}$. If f is given as a CNF formula, then ψ' is a maximal colouring under f if ψ' satisfies the $\mathcal{S}'$-punctuated formula. If f is given as a DNF formula, then ψ' is a minimal colouring under f if ψ' satisfies the $\mathcal{S}'$-punctuated formula.

Example 8.6.4. Consider the 3×3 Hex game. Using the coordinate system introduced in Section 2.6 the outcome function in CNF is:

$$\begin{aligned} \psi \mapsto & (\psi_{a3} \vee \psi_{b3} \vee \psi_{c3}) \wedge (\psi_{a3} \vee \psi_{b3} \vee \psi_{c2}) \wedge (\psi_{a3} \vee \psi_{b2} \vee \psi_{c2}) \wedge (\psi_{a3} \vee \psi_{b2} \vee \psi_{c1}) \wedge \\ & (\psi_{a2} \vee \psi_{b2} \vee \psi_{c2}) \wedge (\psi_{a2} \vee \psi_{b2} \vee \psi_{c1}) \wedge (\psi_{a2} \vee \psi_{b1} \vee \psi_{c1}) \wedge (\psi_{a1} \vee \psi_{b1} \vee \psi_{c1}) \wedge \\ & (\psi_{a2} \vee \psi_{b2} \vee \psi_{b3} \vee \psi_{c3}) \wedge (\psi_{a1} \vee \psi_{b1} \vee \psi_{b2} \vee \psi_{c2}) \wedge (\psi_{a1} \vee \psi_{b1} \vee \psi_{b2} \vee \psi_{b3} \vee \psi_{c3}). \end{aligned}$$

Let $\mathcal{S}' = \{a1, b1\}$, then the $\mathcal{S}'$-punctuated formula is

$$\psi \mapsto (\psi_{b1}) \wedge (\psi_{a1} \vee \psi_{b1}) \wedge (\psi_{a1} \vee \psi_{b1}) \wedge (\psi_{a1} \vee \psi_{b1})$$

which simplifies to $\psi \mapsto \psi_{b1}$. So any colouring ψ' of $\mathcal{S}'$ with ψ'_{b1} = TRUE is a maximal colouring. This means that if MAX wants to move in $\mathcal{S}'$ it would be superrational to set $b1$ to TRUE, since this immediately achieves a maximal colouring. For MIN a superrational move in $\mathcal{S}'$ would be to set $b1$ to FALSE, otherwise MAX can set it to TRUE on the next move and achieve a maximal colouring, reversing MIN's move.

Suppose the element $a2$ is set to TRUE. The outcome function of the Hex game then simplifies to

$$\psi \mapsto (\psi_{a3} \vee \psi_{b3} \vee \psi_{c3}) \wedge (\psi_{a3} \vee \psi_{b3} \vee \psi_{c2}) \wedge (\psi_{a3} \vee \psi_{b2} \vee \psi_{c2}) \wedge (\psi_{a3} \vee \psi_{b2} \vee \psi_{c1}) \wedge \\ (\psi_{a1} \vee \psi_{b1} \vee \psi_{c1}) \wedge (\psi_{a1} \vee \psi_{b1} \vee \psi_{b2} \vee \psi_{c2}) \wedge (\psi_{a1} \vee \psi_{b1} \vee \psi_{b2} \vee \psi_{b3} \vee \psi_{c3}).$$

The $\mathcal{S}'$-punctuated formula is

$$\psi \mapsto (\psi_{a1} \vee \psi_{b1}) \wedge (\psi_{a1} \vee \psi_{b1}) \wedge (\psi_{a1} \vee \psi_{b1})$$

which is simply $\psi \mapsto (\psi_{a1} \vee \psi_{b1})$. So now any colouring ψ' of $\mathcal{S}'$ where just one of the two elements is set to TRUE is a maximal colouring. Since MAX has a second-player strategy that ensures such a maximal colouring is achieved, the set $\mathcal{S}'$ is captured by MAX and can be filled in with a maximal colouring. After doing that, the outcome function has simplified to

$$\psi \mapsto (\psi_{a3} \vee \psi_{b3} \vee \psi_{c3}) \wedge (\psi_{a3} \vee \psi_{b3} \vee \psi_{c2}) \wedge (\psi_{a3} \vee \psi_{b2} \vee \psi_{c2}) \wedge (\psi_{a3} \vee \psi_{b2} \vee \psi_{c1}).$$

The converse, namely that ψ' is *not* a maximal colouring if $f'(\psi')$ = FALSE, does not always hold. It does hold when the formula contains no subsumed clauses:

Theorem 8.6.5. Let $f : \check{X}^{\mathcal{S}} \to \mathbb{B}$, $\mathcal{S}' \subseteq \mathcal{S}$, and $\psi' \in \check{X}^{\mathcal{S}}$. If f is given as an irreducible CNF formula, then ψ' is a maximal colouring under f if and only if ψ' satisfies the $\mathcal{S}'$-punctuated formula. If f is given as an irreducible DNF formula, then ψ' is a minimal colouring under f if and only if ψ' satisfies the $\mathcal{S}'$-punctuated formula.

For recognizing minimal colourings, the same method is used with CNF formulas. A colouring ψ' of a subset $\mathcal{S}'$ of elements is a minimal colouring if deleting all other elements from the CNF formula leaves a formula that is satisfied by ψ'.

It should be noted that a punctuated formula is not the same as a subgame. In fact, a punctuated formula cannot in general be obtained by colouring some elements. Another property of note is that the punctuated formula method does not require the CNF or DNF formula to be in its most reduced form; it also works when there are subsumed clauses.

The punctuated function approach is equally valid for regular SAT problems. If some set of variables has a maximal colouring ψ, then ψ can be assigned safely, since any true assignment will still be true when re-coloured with ψ. This generalizes the notion of a "pure literal" in SAT, which is a literal that occurs only in negated or only in unnegated form. In those cases the punctuated formula becomes simply $\overline{v_\psi}$ or v_ψ. However in most cases the punctuated formula based on just one element will be $v_\psi \wedge \overline{v_\psi}$ which has no satisfying colouring.

f	f'	$f'/\mathfrak{m}$	$f/\mathfrak{m}$	$(f/\mathfrak{m})'$
ψ_v	χ_v	TRUE	TRUE	TRUE
$-\psi_v$	$-\psi_v$	FALSE	FALSE	FALSE
$\psi_v \vee C'$	$\psi_v \vee C'$	TRUE	TRUE	TRUE
$-\psi_v \vee C'$	$-\psi_v \vee C'$	C'	C'	C'
$\psi_v \vee C''$	ψ_v	TRUE	TRUE	TRUE
$-\psi_v \vee C''$	$-\psi_v$	FALSE	C''	TRUE
$\psi_v \vee C' \vee C''$	$\psi_v \vee C'$	TRUE	TRUE	TRUE
$-\psi_v \vee C' \vee C''$	$-\psi_v \vee C'$	C'	$C' \vee C''$	C'

Table 8.7.1: CNF clauses in binary games and subgames punctuated by $\mathcal{S}'$, where $v \in \mathcal{S}'$ and C' and C'' are clauses containing only elements from inside $\mathcal{S}'$ and outside $\mathcal{S}'$, respectively.

8.7 Mutual Recursion

Recall that some set $\mathcal{S}'$ is captured by MAX if $\Gamma^+_{\mathcal{S}'} * \Box$ is a second-player win, and dominated by MAX if $\Gamma^+_{\mathcal{S}'} * \triangle$ is a first-player win. This suggests a mutually recursive relationship, where a set $\mathcal{S}'$ is dominated whenever there is a move in $\mathcal{S}'$ that leaves a captured set, and $\mathcal{S}'$ is captured whenever any opponent's move in $\mathcal{S}'$ leaves a dominated set. As it turns out both these conjectures are only half true.

Theorem 8.7.1. Let $\Gamma = \langle X^{\mathcal{S}}, f \rangle$, $\mathcal{S}' \subseteq \mathcal{S}$, and $\mathfrak{c} \in \mathfrak{C}$. If $\mathcal{S}'$ is dominated by $\mathfrak{c}$ then there exists a move $\mathfrak{m} = \chi^v$ with $v \in \mathcal{S}'$ such that $\mathcal{S}' - v$ is captured by $\mathfrak{c}$ in $\Gamma/\mathfrak{m}$. If $\mathcal{S}'$ is captured by $\mathfrak{c}$ then for all moves $\mathfrak{m} = \chi^v$ with $v \in \mathcal{S}'$ the set $\mathcal{S}' - v$ is dominated by $\mathfrak{c}$ in $\Gamma/\mathfrak{m}$.

Informally, a dominated set contains a move that leaves a captured set, and any move in a captured set leaves a dominated set. However, the reverse is not necessarily true. Even if some move χ^v leaves a captured set $\mathcal{S}'$, then $\mathcal{S}' + v$ still might not be dominated. Similarly, even when every move leaves a dominated set then the original set still might not be captured.

The difference lies in the comparison between games of the types $(\Gamma/\mathfrak{m})^+_{\mathcal{S}''}$ and $(\Gamma^+_{\mathcal{S}'})/\mathfrak{m}$. If these games were equivalent, then the conjectured mutually recursive relationship would indeed be "if and only if". However, we have the following theorem.

Theorem 8.7.2. Let $\Gamma = \langle X^{\mathcal{S}}, f \rangle$ with $\mathcal{S}' \subseteq \mathcal{S}$ and $\mathfrak{m} = \chi^v \in \mathfrak{M}(\mathcal{S})$. Put $\mathcal{S}'' = \mathcal{S}' - v$. Then $(\Gamma/\mathfrak{m})^+_{\mathcal{S}''} \geqslant (\Gamma^+_{\mathcal{S}'})/\mathfrak{m}$ and $(\Gamma/\mathfrak{m})^-_{\mathcal{S}''} \leqslant (\Gamma^-_{\mathcal{S}'})/\mathfrak{m}$. Equality does *not* necessarily hold.

If f is given in CNF, then Table 8.7.1 reveals that the games $(\Gamma/\mathfrak{m})^+_{\mathcal{S}''}$ and $(\Gamma^+_{\mathcal{S}'})/\mathfrak{m}$ diverge when there is a clause of the form $\big((\psi_v \neq \chi) \vee C''\big)$, where C'' is a clause containing no elements from $\mathcal{S}'$.

The failure of the conjecture can be understood intuitively as follows. Assume player $\mathfrak{c}$ attempts to achieve an optimal colouring of $\mathcal{S}'$. After a move $\mathfrak{m} = \chi^v$ is played in $\mathcal{S}'$, the subsequent goal for $\mathfrak{c}$ is to achieve some colouring ψ'' of $\mathcal{S}' - v$ such that $\psi''\mathfrak{m}$ is an optimal colouring of $\mathcal{S}'$ in Γ. For this it is certainly necessary that ψ'' be an optimal colouring of $\mathcal{S}' - v$ in the resulting game $\Gamma/\mathfrak{m}$, but not sufficient, as the addition of $\mathfrak{m}$ might "spoil" the optimality.

What is needed, then, is the assurance that when $\mathfrak{c}$ reaches some optimal colouring ψ'' of $\mathcal{S}' - v$, then $\psi''\mathfrak{m}$ is also an optimal colouring of the same type. This is certainly true in isotone games like Hex, but not in general.

Definition 8.7.3. Let $\Gamma = \langle X^{\mathcal{S}}, f\rangle$, $\mathcal{S}'' \subseteq \mathcal{S}$, $\mathfrak{c} \in \mathfrak{C}$, and $\mathfrak{m} = \chi^v$ with $v \notin \mathcal{S}''$. Then $\mathfrak{m}$ **augments** $\mathcal{S}'$ for player $\mathfrak{c}$ if $\psi''\mathfrak{m}$ is $\mathfrak{c}$-optimal for some $\mathfrak{c}$-optimal $\psi'' \in \check{X}^{\mathcal{S}''}$.

Checking whether a move augments a set is simplified by the following theorem:

Theorem 8.7.4. Let $\Gamma = \langle X^{\mathcal{S}}, f\rangle$, $\mathcal{S}'' \subseteq \mathcal{S}$, $\mathfrak{c} \in \mathfrak{C}$ and $\mathfrak{m} = \chi^v$ with $v \notin \mathcal{S}''$. If $\psi''\mathfrak{m}$ is $\mathfrak{c}$-optimal for some $\mathfrak{c}$-optimal $\psi'' \in \check{X}^{\mathcal{S}''}$, then $\psi''\mathfrak{m}$ is $\mathfrak{c}$-optimal for every $\mathfrak{c}$-optimal $\psi'' \in \check{X}^{\mathcal{S}''}$.

> *Corollary i:* If $\psi''\mathfrak{m}$ is not $\mathfrak{c}$-optimal for some $\mathfrak{c}$-optimal $\psi'' \in \check{X}^{\mathcal{S}''}$, then $\psi''\mathfrak{m}$ is not $\mathfrak{c}$-optimal for any $\mathfrak{c}$-optimal $\psi'' \in \check{X}^{\mathcal{S}''}$.

According to this theorem, the requirement "for some" in Definition 8.7.3 can equivalently be replaced with "for all". The mutually recursive relationship between captured and dominated sets is then given as follows.

Theorem 8.7.5. Let $\Gamma = \langle X^{\mathcal{S}}, f\rangle$, $\mathcal{S}' \subseteq \mathcal{S}$, and $\mathfrak{c} \in \mathfrak{C}$. If for all moves $\mathfrak{m} = \chi^v \in \mathfrak{M}(\mathcal{S}')$ the set $\mathcal{S}' - v$ is $\mathfrak{c}$-dominated in $\Gamma/\mathfrak{m}$, and $\mathfrak{m}$ $\mathfrak{c}$-augments $\mathcal{S}'$, then $\mathcal{S}'$ is $\mathfrak{c}$-captured. If there exists a move $\mathfrak{m} = \chi^v \in \mathfrak{M}(\mathcal{S}')$ such that $\mathcal{S}' - v$ is $\mathfrak{c}$-captured in $\Gamma/\mathfrak{m}$, and $\mathfrak{m}$ $\mathfrak{c}$-augments $\mathcal{S}'$,b then $\mathcal{S}'$ is $\mathfrak{c}$-dominated and $\mathfrak{m}$ is a $\mathfrak{c}$-dominating move in $\mathcal{S}'$.

One can view the augmentation requirement for captured sets as stating that $\mathfrak{c}$ ends up reversing move $\mathfrak{m}$. A few obvious cases of augmentation are:

- If $\mathfrak{m}$ is $\mathfrak{c}$-rational then it $\mathfrak{c}$-augments any set $\mathcal{S}'$. This means that a set is $\mathfrak{c}$-dominated *if and only if* there is a rational move that leaves a captured set.
- If v is dead in Γ/ψ'' for some $\mathfrak{c}$-optimal colouring of $\mathcal{S}'$, then any move $\chi^{\mathfrak{m}}$ will $\mathfrak{c}$-augment $\mathcal{S}'$. This means that a set is $\mathfrak{c}$-captured if every move leaves a $\mathfrak{c}$-dominated set which, after substitution, kills the move.
- If $\mathfrak{m}$ is $\mathfrak{c}$-preferable to $\mathfrak{m}'$, and $\mathfrak{m}'$ $\mathfrak{c}$-augments some set, then $\mathfrak{m}$ $\mathfrak{c}$-augments the same set. This is true even if $\mathfrak{m}$ and $\mathfrak{m}'$ do not colour the same element.

With these observations, the mutually recursive rules for isotone games are:

- A set $\mathcal{S}'$ is $\mathfrak{c}$-dominated if and only if there is a rational move in $\mathcal{S}'$ that leaves a $\mathfrak{c}$-captured set.
- A set $\mathcal{S}'$ is $\mathfrak{c}$-captured if and only if any $\bar{\mathfrak{c}}$-rational move $\mathfrak{m}$ in $\mathcal{S}'$ leaves a $\mathfrak{c}$-dominated set which, after substitution with $\mathfrak{c}$-rational moves, kills $\mathfrak{m}$.

The bottom of the mutual recursion is provided by the following rules. Let $\Gamma = \langle X^{\mathcal{S}}, f\rangle$, $\mathcal{S}' \subseteq \mathcal{S}$, and $\mathfrak{c} \in \mathfrak{C}$, then:

- If $\mathcal{S}' = \varnothing$ then $\mathcal{S}'$ is $\mathfrak{c}$-captured.
- If $\mathcal{S}' = \{v\}$, so that $|\mathcal{S}'| = 1$, then $\mathcal{S}'$ is $\mathfrak{c}$-dominated if and only if there is a $\mathfrak{c}$-rational move χ^v.
- If all elements in $\mathcal{S}'$ are dead then $\mathcal{S}'$ is $\mathfrak{c}$-captured.

There rules are easily verified directly from the definitions.

8.8 Superrational Play

The existence of captured and dominated sets leads to the strategic advice of **superrational play**. This advice is as follows:

1. If there exists a captured subset, then perform the associated substitution.
2. If there exists a dominated subset, then only one move from this set needs to be considered, provided that the move to be considered is a dominating move.

The term *superrational* indicates that this strategy is a generalization of the theory of rational moves from Section 5.2. A rational move χ^v is in fact a dominating move in the subset $\{v\}$, and a dead cell is one that is captured by both players, so that any colouring of that cell is a legal substitution.

The substitution strategy is justified immediately by Theorem 8.4.2, as a substitution does not change the minimax value of the position. It does simplify the analysis of the position, in that there are then fewer moves to consider. Note that the game after substitution has acquired a star if the captured set was odd sized and the game is not isotone. In the original game, without the substitution, the star move is represented by any random move in the captured set. Both players can essentially treat the captured set as a repository of star moves, and one only needs to move in a captured set if a star move is required in the substituted game. In such a case, the captor needs to make sure that the move does indeed capture the set, whereas the other player can choose any random move within the captured set.

The domination strategy comes from Theorem 8.5.2, as remarked previously: A dominating move in a subset is guaranteed to be at least as good as any other move in that subset. It is very well possible that a dominated set contains more than one dominating move. For this reason, whenever a dominating move is found one cannot simply eliminate all of its dominated moves from consideration, because in a set with more than one dominating move that would eliminate all moves.

One can harmonize the superrational strategies by defining a ternary local game. Given a game $\langle X^{\mathcal{S}}, f\rangle$ and some subset $\mathcal{S}' \subseteq \mathcal{S}$, one can consider the subgame in which MAX attempts to attain a maximal colouring and MIN attempts to attain a minimal colouring of $\mathcal{S}'$. If neither goal is achieved, the local game is declared a draw. The games $\Gamma^+_{\mathcal{S}'}$ and $\Gamma^-_{\mathcal{S}'}$ are the two binary variants of this ternary game, namely the ones where the draw outcome has been declared a win for MIN and for MAX, respectively. As a consequence, is can be shown that punctuated formulas have the property that under any colouring the value of the CNF-punctuated formula is always at least equal to the value of the DNF-punctuated formula.

Considering this ternary local game, the superrational play strategy can be re-worded as "do not make any local mistakes". More explicitly:

- If the local ternary game is a win, then only one locally winning move needs to be examined.
- If the local ternary game is a draw, then only all the locally drawing moves need to be examined.
- If the local ternary game is a loss, then the subset is already captured by the opponent.

The justification is that if a given move $\mathfrak{m}$ wins the local ternary game, then it is a dominating move and therefore at least as good in Γ as any other move in $\mathcal{S}'$. If $\mathfrak{m}$ loses the local ternary game, then it leaves a dominated set for the opponent, and $\mathfrak{m}$ apparently augments the remaining $\mathcal{S}' \setminus \mathfrak{m}$ for the opponent, as it is part of the eventual optimal colouring that the opponent can reach. Another way of saying this is that a locally losing move can be reversed by an appropriate local reply. Therefore any move in the local ternary game was at least as good as $\mathfrak{m}$ in Γ, so if there are locally non-losing moves then the locally losing moves can be ignored.

The superrational strategy does not give any advice as to whether or not it is wise to move in a certain subset $\mathcal{S}'$ in the first place. It merely says that *if* one wants to move in $\mathcal{S}'$, *then* one must take care not to make a local mistake.

8.9 Proofs

Theorem 8.1.3. Let $\Gamma = \langle X^{\mathcal{S}}, f \rangle$ and $\psi_0, \psi_1 \in X^{\mathcal{S}}$. If ψ_0 and ψ_1 are both maximal colourings then $\psi_0\psi_1$ is also a maximal colouring. If ψ_0 and ψ_1 are both minimal colourings then $\psi_0\psi_1$ is also a minimal colouring.

Proof. If both ψ_0 and ψ_1 are maximal colourings then for any pure colouring $\psi^* \in \check{X}^{\mathcal{S}}$ we have $f\big(\psi^*(\psi_0\psi_1)\big) = f\big((\psi^*\psi_0)\psi_1\big) \geqslant f(\psi^*\psi_0) \geqslant f(\psi^*)$. If ψ_0 and ψ_1 are minimal colourings then similarly $f\big(\psi^*(\psi_0\psi_1)\big) \leqslant f(\psi^*)$. □

Theorem 8.2.3. Let $\Gamma = \langle X^{\mathcal{S}}, f \rangle$ and $\psi \in X^{\mathcal{S}}$. Then

$$\Gamma^+_\psi \wedge \Gamma/\psi \leqslant \Gamma,$$

$$\Gamma^-_\psi \vee \Gamma/\psi \geqslant \Gamma.$$

Proof. Note that $\Gamma^+_\psi \wedge \Gamma/\psi$ is played on $X^{\mathcal{S}}$. The proof requires that any $\psi^* \in \check{X}^{\mathcal{S}}$ that is a win for MAX on $\Gamma^+_\psi \wedge \Gamma/\psi$ is also a win for MAX on Γ. Let $\psi^* \in \check{X}^{\mathcal{S}}$ be a win on $\Gamma^+_\psi \wedge \Gamma/\psi$. Then ψ^* is a win on $\Gamma^+_\psi * \mathcal{S}$ and on $\Gamma/\psi * \mathcal{S}$. Put $\psi^*_{\mathcal{A}} = \psi^* \searrow \mathcal{A}(\psi)$ and $\psi^*_{\mathcal{U}} = \psi^* \searrow \mathcal{U}(\psi)$, so that $\psi^*_{\mathcal{U}}\psi^*_{\mathcal{A}} = \psi^*$ due to the partitioning of $\mathcal{D}(\psi^*)$ into $\mathcal{A}(\psi)$ and $\mathcal{U}(\psi)$. Since Γ^+_ψ is played on $X^{\mathcal{A}(\psi)}$, we have that

$\psi^*_{\mathcal{A}}$ is a win on Γ^+_ψ, and therefore $\Gamma/\psi^*_{\mathcal{A}} \geqslant \Gamma/\psi$ by Definition 8.2.1. Similarly, since Γ/ψ is played on $X^{\mathcal{U}(\psi)}$, we have that $\psi^*_{\mathcal{U}}$ is a win on Γ/ψ, and therefore $f(\psi^*_{\mathcal{U}}\psi) = +1$ by Definition 6.1.1. Now observe that

$$\psi^*_{\mathcal{U}}\psi\psi^*_{\mathcal{A}} = \psi^*_{\mathcal{U}}(\psi\psi^*_{\mathcal{A}}) = \psi^*_{\mathcal{U}}\psi^*_{\mathcal{A}} = \psi^*$$

from Observation 2.2.1:ii since $\mathcal{A}(\psi^*_{\mathcal{A}}) = \mathcal{A}(\psi)$. We then obtain

$$f(\psi^*) = f(\psi^*_{\mathcal{U}}\psi\psi^*_{\mathcal{A}}) \geqslant f(\psi^*_{\mathcal{U}}\psi\psi) = f(\psi^*_{\mathcal{U}}\psi) = +1$$

where the inequality follows from $\Gamma/\psi^*_{\mathcal{A}} \geqslant \Gamma/\psi$. This proves $\Gamma^+_\psi \wedge \Gamma/\psi \leqslant \Gamma$. The proof for $\Gamma^-_\psi \vee \Gamma/\psi \geqslant \Gamma$ is analogous, featuring $\Gamma/\psi^*_{\mathcal{A}} \leqslant \Gamma/\psi$ and $f(\psi^*_{\mathcal{U}}\psi) = -1$ and culminating in $f(\psi^*) \leqslant f(\psi^*_{\mathcal{U}}\psi) = -1$. □

Theorem 8.3.2. Let $\Gamma = \langle X^{\mathcal{S}}, f\rangle$, let $\psi \in X^{\mathcal{S}}$, and let $\mathfrak{c} \in \mathfrak{C}$. If ψ is a maximal colouring then $\text{MNX}(\Gamma/\psi * \mathcal{S}; \mathfrak{c}) \geqslant \text{MNX}(\Gamma; \mathfrak{c})$. If ψ is a minimal colouring then $\text{MNX}(\Gamma/\psi * \mathcal{S}; \mathfrak{c}) \leqslant \text{MNX}(\Gamma; \mathfrak{c})$.

> *Corollary i:* Substituting a maximal colouring cannot hurt MAX, and substituting a minimal colouring cannot hurt MIN.

Proof. If $|\mathcal{A}(\psi)|$ is even and ψ is a maximal colouring, then $\Gamma/\psi \geqslant \Gamma$ by Observation 8.1.2:i, which means $\Gamma/\psi * \mathcal{S} \geqslant \Gamma$ by Definition 6.3.2. Since Γ/ψ is played on $\mathcal{U}(\psi)$ and $|\mathcal{S} \setminus \mathcal{U}(\psi)| = |\mathcal{A}(\psi)|$ is even, by Theorem 6.3.3 we have $\text{MNX}(\Gamma/\psi; \mathfrak{c}) = \text{MNX}(\Gamma/\psi * \mathcal{S}; \mathfrak{c}) \geqslant \text{MNX}(\Gamma; \mathfrak{c})$. If ψ is a minimal colouring then $\text{MNX}(\Gamma/\psi; \mathfrak{c}) = \text{MNX}(\Gamma/\psi * \mathcal{S}; \mathfrak{c}) \leqslant \text{MNX}(\Gamma; \mathfrak{c})$ for the same reasons. If $|\mathcal{A}(\psi)|$ is odd, then add a star to both Γ/ψ and to the remainder of the game, to obtain $\text{MNX}(\Gamma/\psi *; \mathfrak{c}) = \text{MNX}(((\Gamma/\psi)*) * (\mathcal{S} * *); \mathfrak{c}) \geqslant \text{MNX}(\Gamma * *; \mathfrak{c}) = \text{MNX}(\Gamma; \mathfrak{c})$ for a maximal colouring and vice versa for a minimal colouring. □

Theorem 8.4.2. Let $\Gamma = \langle X^{\mathcal{S}}, f\rangle$ and $\mathcal{S}' \subseteq \mathcal{S}$, where $\mathcal{S}'$ is captured by player $\mathfrak{c} \in \mathfrak{C}$ with associated optimal colouring $\psi \in \check{X}^{\mathcal{S}'}$. Then $\text{MNX}(\Gamma * \Box; \bar{\mathfrak{c}}) = \text{MNX}(\Gamma/\psi * \Box; \bar{\mathfrak{c}})$, and $\text{MNX}(\Gamma * \triangle; \mathfrak{c}) = \text{MNX}(\Gamma/\psi * \triangle; \mathfrak{c})$.

Proof. Without loss of generality assume that $\mathfrak{c} = \text{MAX}$, so ψ is a maximal colouring. By Observation 8.1.2:i we have $\Gamma/\psi * \Diamond = (\Gamma * \Diamond)/\psi \geqslant \Gamma * \Diamond$ for any $\Diamond \in \{\Box, \triangle\}$, and by Observation 6.3.1:i this means $\text{MNX}(\Gamma/\psi * \Diamond; \mathfrak{c}') \geqslant \text{MNX}(\Gamma * \Diamond; \mathfrak{c}')$ for any $\mathfrak{c}' \in \mathfrak{C}$. What remains to be proved is $\text{MNX}(\Gamma/\psi * \Box; \text{MIN}) \leqslant \text{MNX}(\Gamma * \Box; \text{MIN})$ and $\text{MNX}(\Gamma/\psi * \triangle; \text{MAX}) \leqslant \text{MNX}(\Gamma * \triangle; \text{MAX})$. This is equivalent to $\text{MNX}(\Gamma/\psi * \Box; \text{MIN}) = +1 \Longrightarrow \text{MNX}(\Gamma * \Box; \text{MIN}) = +1$ and $\text{MNX}(\Gamma/\psi * \triangle; \text{MAX}) = +1 \Longrightarrow \text{MNX}(\Gamma * \triangle; \text{MAX}) = +1$. Since $\mathcal{S}'$ was captured we have, by Definition 8.4.1, $\text{MNX}(\Gamma^+_\psi; \text{MIN}) = +1$. The two implications then follow from Theorems 6.4.2 and 6.4.4, respectively. □

Theorem 8.5.2. Let $\Gamma = \langle X^{\mathcal{S}}, f\rangle$, and let $\mathcal{S}'$ be dominated by $\mathfrak{c} \in \mathfrak{C}$ with associated optimal colouring $\psi \in \check{X}^{\mathcal{S}'}$ and dominating move $\mathfrak{m}$. Let $\mathfrak{m}' \in \mathfrak{M}(\mathcal{S}')$. Then $\text{NGX}((\Gamma * \triangle)/\mathfrak{m}; \bar{\mathfrak{c}}) \leqslant \text{NGX}((\Gamma * \triangle)/\mathfrak{m}'; \bar{\mathfrak{c}})$.

Proof. Without loss of generality assume that $\mathfrak{c} = \text{MAX}$, so ψ is a maximal colouring. To prove is $\text{NGX}((\Gamma * \triangle)/\mathfrak{m}; \text{MIN}) \leqslant \text{NGX}((\Gamma * \triangle)/\mathfrak{m}'; \text{MIN})$, for which it suffices to prove $\text{MNX}((\Gamma * \triangle)/\mathfrak{m}; \text{MIN}) = -1 \Longrightarrow \text{MNX}((\Gamma * \triangle)/\mathfrak{m}'; \text{MIN}) = -1$. Since $\mathfrak{m}$ was a dominating move in $\Gamma^+_{\mathcal{S}'}$ we have $\text{MNX}(\Gamma^+_{\mathcal{S}'}/\mathfrak{m}; \text{MIN}) = +1$. Note that $(\Gamma * \triangle)/\mathfrak{m}$ is even. If $\text{MNX}((\Gamma * \triangle)/\mathfrak{m}; \text{MIN}) = -1$ then apparently $\text{MNX}(((\Gamma * \triangle)/\mathfrak{m})/\psi; \text{MIN}) = -1$, otherwise Theorem 6.4.4 would imply that $\text{MNX}((\Gamma * \triangle)/\mathfrak{m}; \text{MIN}) = +1$.

Since ψ is a maximal colouring we then have $\text{MNX}(((\Gamma*\triangle)/\mathfrak{m}')/\psi; \text{MIN}) = \text{MNX}((\Gamma*\triangle)/\psi\mathfrak{m}'; \text{MIN}) \leqslant \text{MNX}((\Gamma*\triangle)/\psi; \text{MIN}) = -1$ and therefore MIN has a winning partition strategy in $(\Gamma*\triangle)/\mathfrak{m}'$ and therefore $\text{MNX}((\Gamma*\triangle)/\mathfrak{m}'; \text{MIN}) = -1$. □

Theorem 8.6.3. Let $f : \check{X}^{\mathcal{S}} \to \mathbb{B}$, $\mathcal{S}' \subseteq \mathcal{S}$, and $\psi' \in \check{X}^{\mathcal{S}'}$. If f is given as a CNF formula, then ψ' is a maximal colouring under f if ψ' satisfies the $\mathcal{S}'$-punctuated formula. If f is given as a DNF formula, then ψ' is a minimal colouring under f if ψ' satisfies the $\mathcal{S}'$-punctuated formula.

Proof. Assume f is written as the CNF formula $\mathcal{C}_0 \wedge \mathcal{C}_1 \wedge \ldots \wedge \mathcal{C}_{n-1}$. Let ψ' satisfy the $\mathcal{S}'$-punctuated formula. To prove is that for any $\xi \in X^{\mathcal{S}}$ we have $f(\xi\psi') \geqslant f(\xi)$, which is equivalent with $f(\xi) = +1 \Longrightarrow f(\xi\psi') = +1$. Let $\xi \in X^{\mathcal{S}}$ with $f(\xi) = +1$. Consider some clause $\mathcal{C}_i$ from the CNF formula. Since $f(\xi) = +1$ we have $\mathcal{C}_i(\xi) = +1$. Now distinguish two cases, based on whether $\mathcal{C}_i$ contains any elements from $\mathcal{S}'$.

Case: $\mathcal{C}_i$ contains elements from $\mathcal{S}'$. Put $\mathcal{C}_i = \mathcal{C}_i' \vee \mathcal{C}_i''$ where $\mathcal{C}_i'$ contains only elements from $\mathcal{S}'$ and $\mathcal{C}_i''$ contains no elements from $\mathcal{S}'$. The punctuated formula contains the clause $\mathcal{C}_i'$, and therefore $\mathcal{C}_i'(\psi') = +1$ since ψ' satisfies the punctuated formula. As $\mathcal{C}_i'$ contains only elements from $\mathcal{A}(\psi')$ we have $\mathcal{C}_i'(\xi\psi') = \mathcal{C}_i'(\psi') = +1$ and therefore $\mathcal{C}_i(\xi\psi') = \mathcal{C}_i'(\xi\psi') \vee \mathcal{C}_i''(\xi\psi') = +1$.

Case: $\mathcal{C}_i$ contains no elements from $\mathcal{S}'$. Then $\mathcal{C}_i(\xi\psi') = \mathcal{C}_i(\xi)$, and since ξ satisfies the whole formula, this equals $+1$.

In either case we have $\mathcal{C}_i(\xi\psi') = +1$. Therefore $\xi\psi'$ satisfies all clauses, and $f(\xi\psi') = +1$. The proof for minimal colourings and DNF formula is analogous. □

Theorem 8.6.5. Let $f : \check{X}^{\mathcal{S}} \to \mathbb{B}$, $\mathcal{S}' \subseteq \mathcal{S}$, and $\psi' \in \check{X}^{\mathcal{S}}$. If f is given as an irreducible CNF formula, then ψ' is a maximal colouring under f if and only if ψ' satisfies the $\mathcal{S}'$-punctuated formula. If f is given as an irreducible DNF formula, then ψ' is a minimal colouring under f if and only if ψ' satisfies the $\mathcal{S}'$-punctuated formula.

Proof. Consider the CNF case, and denote the $\mathcal{S}'$-punctuated formula as f'. . From Theorem 8.6.3 it is already known that ψ' is a maximal colouring under f if $f'(\psi') = +1$. Suppose now that $f'(\psi') = -1$, and that in particular it does not satisfy the first k clauses of f', so that $\mathcal{C}_i'(\psi') = -1$ for $i < k$ and $\mathcal{C}_i'(\psi') = +1$ for $i \geqslant k$, with $k \geqslant 1$. For ψ' not to be a maximal colouring there would have to exist some $\psi^* \in \check{X}^{\mathcal{S}}$ with $f(\psi^*) = +1$ and $f(\psi^*\psi') = -1$. That means that at least one of the clauses $\mathcal{C}_0 \wedge \ldots \wedge \mathcal{C}_{k-1}$ is flipped to FALSE by the re-colouring $\psi^*\psi'$, so there exists some $i < k$ with $\mathcal{C}_i'(\psi') = -1$, $\mathcal{C}_i''(\psi^*) = -1$, and $\mathcal{C}_i(\psi^*) = \mathcal{C}_i'(\psi^*) \vee \mathcal{C}_i''(\psi^*) = +1$ so $\mathcal{C}_i'(\psi^*) = +1$. Such a ψ^* exists, unless $f(\psi^*) = +1$ implies $\mathcal{C}_0'(\psi^*) \wedge \ldots \wedge \mathcal{C}_{k-1}'(\psi^*)$.

If $f(\psi^*) = +1$ does imply $\mathcal{C}_0'(\psi^*) \wedge \ldots \wedge \mathcal{C}_{k-1}'(\psi^*)$, then $\mathcal{C}_0'', \mathcal{C}_1'', \ldots, \mathcal{C}_{k-1}''$ were "superfluous" and can all be deleted from the formula without changing the outcome. In other words, the clauses $\mathcal{C}_0, \mathcal{C}_1, \ldots, \mathcal{C}_{k-1}$ were reducible. The conclusion then is that $f(\psi') = +1$ is a necessary condition for ψ' being a maximal colouring if all clauses are irreducible. □

Theorem 8.7.1. Let $\Gamma = \langle X^{\mathcal{S}}, f\rangle$, $\mathcal{S}' \subseteq \mathcal{S}$, and $\mathfrak{c} \in \mathfrak{C}$. If $\mathcal{S}'$ is dominated by $\mathfrak{c}$ then there exists a move $\mathfrak{m} = \chi^v$ with $v \in \mathcal{S}'$ such that $\mathcal{S}' - v$ is captured by $\mathfrak{c}$ in $\Gamma/\mathfrak{m}$. If $\mathcal{S}'$ is captured by $\mathfrak{c}$ then for all moves $\mathfrak{m} = \chi^v$ with $v \in \mathcal{S}'$ the set $\mathcal{S}' - v$ is dominated by $\mathfrak{c}$ in $\Gamma/\mathfrak{m}$.

Proof. This follows immediately from Definitions 8.5.1 and 8.4.1 with definition 5.1.1. □

Theorem 8.7.2. Let $\Gamma = \langle X^{\mathcal{S}}, f\rangle$ with $\mathcal{S}' \subseteq \mathcal{S}$ and $\mathfrak{m} = \chi^v \in \mathfrak{M}(\mathcal{S})$. Put $\mathcal{S}'' = \mathcal{S}' - v$. Then $(\Gamma/\mathfrak{m})^+_{\mathcal{S}''} \geqslant (\Gamma^+_{\mathcal{S}'})/\mathfrak{m}$ and $(\Gamma/\mathfrak{m})^-_{\mathcal{S}''} \leqslant (\Gamma^-_{\mathcal{S}'})/\mathfrak{m}$. Equality does *not* necessarily hold.

Proof. Writing out the game definitions we obtain

$$(\Gamma/\mathfrak{m})^+_{\mathcal{S}''} = \left\langle X^{\mathcal{S}\setminus v}, \xi \mapsto \begin{cases} +1 & \text{if } \forall_{\psi^* \in \check{X}^{\mathcal{S}}}[f(\psi^*\xi\mathfrak{m}) \geqslant f(\psi^*\mathfrak{m})], \\ -1 & \text{otherwise} \end{cases}\right\rangle$$

and

$$(\Gamma^+_{\mathcal{S}'})/\mathfrak{m} = \left\langle X^{\mathcal{S}\setminus v}, \xi \mapsto \begin{cases} +1 & \text{if } \forall_{\psi^* \in \check{X}^{\mathcal{S}}}[f(\psi^*\xi\mathfrak{m}) \geqslant f(\psi^*)], \\ -1 & \text{otherwise.} \end{cases}\right\rangle$$

Let $\xi \in \check{X}^{\mathcal{S}\setminus v}$, then it needs to be shown that if ξ wins for MAX in $(\Gamma^+_{\mathcal{S}'})/\mathfrak{m}$ then ξ also wins for MAX in $(\Gamma/\mathfrak{m})^+_{\mathcal{S}''}$. Let $\psi^* \in \check{X}^{\mathcal{S}}$. If $f(\psi^*\xi\mathfrak{m}) \geqslant f(\psi^*)$ then $f(\psi^*\xi\mathfrak{m}) = f(\psi^*\mathfrak{m}\xi\mathfrak{m}) \geqslant f(\psi^*\mathfrak{m})$. The proof for $(\Gamma/\mathfrak{m})^-_{\mathcal{S}''} \leqslant (\Gamma^-_{\mathcal{S}'})/\mathfrak{m}$ is entirely analogous. A counterexample for equality was already given in Table 8.7.1 and accompanying text in Section 8.7. □

Theorem 8.7.4. Let $\Gamma = \langle X^{\mathcal{S}}, f\rangle$, $\mathcal{S}'' \subseteq \mathcal{S}$, $\mathfrak{c} \in \mathfrak{C}$ and $\mathfrak{m} = \chi^v$ with $v \notin \mathcal{S}''$. If $\psi''\mathfrak{m}$ is $\mathfrak{c}$-optimal for some $\mathfrak{c}$-optimal $\psi'' \in \check{X}^{\mathcal{S}''}$, then $\psi''\mathfrak{m}$ is $\mathfrak{c}$-optimal for every $\mathfrak{c}$-optimal $\psi'' \in \check{X}^{\mathcal{S}''}$.

> *Corollary i:* If $\psi''\mathfrak{m}$ is not $\mathfrak{c}$-optimal for some $\mathfrak{c}$-optimal $\psi'' \in \check{X}^{\mathcal{S}''}$, then $\psi''\mathfrak{m}$ is not $\mathfrak{c}$-optimal for any $\mathfrak{c}$-optimal $\psi'' \in \check{X}^{\mathcal{S}''}$.

Proof. Without loss of generality let $\mathfrak{c}$ = MAX. Let $\psi''_1, \psi''_2 \in \check{X}^{\mathcal{S}''}$ be maximal colourings where $\psi''_1\mathfrak{m}$ is also a maximal colouring. Note that $\psi''_2\mathfrak{m} = \mathfrak{m}\psi''_2$ as $\mathcal{A}(\mathfrak{m}) = v \notin \mathcal{S}'' = \mathcal{A}(\psi''_2)$, and that $\psi''_1\psi''_2 = \psi''_2$ as $\mathcal{A}(\psi''_1) = \mathcal{S}'' = \mathcal{A}(\psi''_2)$. For any $\xi \in \check{X}^{\mathcal{S}}$ we then have $f(\xi\psi''_2\mathfrak{m}) = f(\xi\psi''_1\psi''_2\mathfrak{m}) = f(\xi\psi''_1\mathfrak{m}\psi''_2) \geqslant f(\xi\psi''_1\mathfrak{m}) \geqslant f(\xi)$, so $\psi''_2\mathfrak{m}$ is also a maximal colouring. □

Theorem 8.7.5. Let $\Gamma = \langle X^{\mathcal{S}}, f\rangle$, $\mathcal{S}' \subseteq \mathcal{S}$, and $\mathfrak{c} \in \mathfrak{C}$. If for all moves $\mathfrak{m} = \chi^v \in \mathfrak{M}(\mathcal{S}')$ the set $\mathcal{S}' - v$ is $\mathfrak{c}$-dominated in $\Gamma/\mathfrak{m}$, and $\mathfrak{m}$ $\mathfrak{c}$-augments $\mathcal{S}'$, then $\mathcal{S}'$ is $\mathfrak{c}$-captured. If there exists a move $\mathfrak{m} = \chi^v \in \mathfrak{M}(\mathcal{S}')$ such that $\mathcal{S}' - v$ is $\mathfrak{c}$-captured in $\Gamma/\mathfrak{m}$, and $\mathfrak{m}$ $\mathfrak{c}$-augments $\mathcal{S}'$,b then $\mathcal{S}'$ is $\mathfrak{c}$-dominated and $\mathfrak{m}$ is a $\mathfrak{c}$-dominating move in $\mathcal{S}'$.

Proof. Without loss of generality let $\mathfrak{c}$ = MAX. Suppose that for all moves $\mathfrak{m} = \chi^v \in \mathfrak{M}(\mathcal{S}')$ the set $\mathcal{S}' - v$ is MAX-dominated in $\Gamma/\mathfrak{m}$, and $\mathfrak{m}$ MAX-augments $\mathcal{S}'$. Let ψ' be some maximal colouring of $\mathcal{S}'$, and let $\mathfrak{m} = \chi^v \in \mathfrak{M}(\mathcal{S}')$. Then $\text{MNX}((\Gamma/\mathfrak{m})^+_{\psi''} * \triangle; \text{MAX}) = +1$ for some maximal colouring ψ'' of $\mathcal{S}' - v$, so $\text{MNX}(\Gamma^+_{\psi''\mathfrak{m}}/\mathfrak{m} * \triangle; \text{MAX}) = +1$. Since $\mathfrak{m}$ MAX-augments $\mathcal{S}' - v$ we have that $\psi''\mathfrak{m}$ is a maximal colouring of $\mathcal{S}'$, and therefore $\Gamma^+_{\psi'} = \Gamma^+_{\psi''\mathfrak{m}}$. So then $\text{MNX}(\Gamma^+_{\psi'}/\mathfrak{m} * \triangle; \text{MAX}) = +1$ for any $\mathfrak{m} \in \mathfrak{M}(\mathcal{S}')$, which by Observation 5.1.3:iii means $\text{MNX}(\Gamma^+_{\psi'} * \square; \text{MIN}) = +1$, so by Definition 8.4.1 this means that $\mathcal{S}'$ is MAX-captured.

Suppose there exists a move $\mathfrak{m} = \chi^v \in \mathfrak{M}(\mathcal{S}')$ such that $\mathcal{S}' - v$ is MAX-captured in $\Gamma/\mathfrak{m}$, and $\mathfrak{m}$ MAX-augments $\mathcal{S}'$. Then by the same reasoning we have $\text{MNX}(\Gamma^+_{\psi'}/\mathfrak{m} * \square; \text{MIN}) = +1$ for some maximal colouring ψ' of $\mathcal{S}'$. From Observation 5.1.3:ii this implies that MNX($\Gamma^{+}_{\psi'} * \triangle$; MAX) $= +1$, so by Definition 8.5.1 this means that $\mathcal{S}'$ is MAX-dominated, and that in fact $\mathfrak{m}$ is a $\mathcal{S}'$-dominating move for MAX. □

Chapter 9

Dynamic Traces

The previous chapter studies subgames whose goal is to obtain an optimal colouring ψ of a set of elements. A special case occurs when ψ is sufficient to settle the value of the outcome function. Such a colouring is of course optimal, and winning the embedded component Γ_ψ^+ or Γ_ψ^- can be sufficient for the appropriate player to win the full game. In general, it is possible to define embedded games that are sufficient to win the overall game, and to discover such games dynamically.

Techniques like these are also used in the game of Go, where a "trace" keeps track of the set of board locations actually used in achieving some goal. Several Go program authors use this approach in various forms, though nothing has apparently been published about the subject. The idea of a trace is that it is some subset of the board having a property that can be proved irrespective of the status of the board outside of the trace.

9.1 Winning Embedded Components

In Section 6.3 the concept of necessary and sufficient games was defined for games played on the same colour space. The definition is in terms of the partial order comparison for games, which was then expanded to games that are not played on the same colour space. Combining this, we arrive at the notion of a dynamic trace.

Informally, a dynamic trace is a smaller game that one of the players can win, and that is sufficient for the same player to win the larger game. This player can thus win the larger game by concentrating only on the smaller game. Two examples for Hex are displayed in Figure 9.1. Seasoned Hex players will notice that only the marked empty cells are relevant in the positions shown. These sets of marked cells form an "explanation" of sorts for the victory.

The dynamic trace concept will be made precise in the following definitions. The definitions only

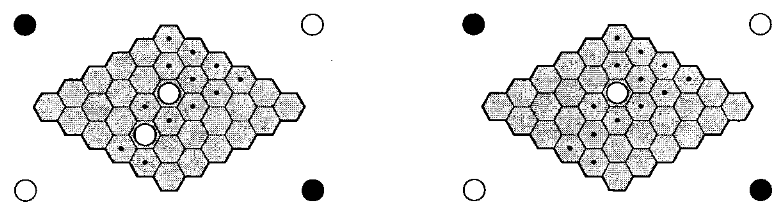


Figure 9.1: Dynamic traces for White, with Black (left) and White (right) to move.

relate to initial positions, and as with other definitions of this nature they are extended to other positions by considering the subgame induced by the position.

Definition 9.1.1. Let $\Gamma = \langle X^{\mathcal{S}}, f \rangle$ and $\Gamma' = \langle X^{\mathcal{S}'}, f' \rangle$ for some $\mathcal{S}' \subseteq \mathcal{S}$. Let $\mathfrak{c} \in \mathfrak{C}$. Then Γ' is **a dynamic trace** for MAX in $(\Gamma, \mathfrak{c})$ if the following two conditions are satisfied:

- $\Gamma' \leqslant \Gamma$;
- $\text{MNX}(\Gamma' * \mathcal{S}; \mathfrak{c}) = +1$.

Similarly, Γ' is a dynamic trace for MIN in $(\Gamma, \mathfrak{c})$ if $\Gamma' \geqslant \Gamma$ and $\text{MNX}(\Gamma' * \mathcal{S}; \mathfrak{c}) = -1$.

> *Observation i:* If there exists a dynamic trace for MAX in $(\Gamma, \mathfrak{c})$ then $\text{MNX}(\Gamma; \mathfrak{c}) = +1$, and if there exists one for MIN then $\text{MNX}(\Gamma; \mathfrak{c}) = -1$. This is Observation 6.3.1:i, which applies since $\Gamma' \leqslant \Gamma$ by definition in this case means $\Gamma' * \mathcal{S} \leqslant \Gamma$.
>
> *Observation ii:* The game Γ itself is a dynamic trace in $(\Gamma, \mathfrak{c})$, for MAX if $\text{MNX}(\Gamma; \mathfrak{c}) = +1$ and for MIN if $\text{MNX}(\Gamma; \mathfrak{c}) = -1$. This is because $\Gamma * \mathcal{S} = \Gamma$.
>
> *Observation iii:* From Observation i it follows that at most one player can have a dynamic trace in $(\Gamma, \mathfrak{c})$, and from Observation ii it follows that at least one player will have one. So *exactly* one player will have dynamic traces in any game $(\Gamma, \mathfrak{c})$.
>
> *Observation iv:* If Γ is isotone then the term $\text{MNX}(\Gamma' * \mathcal{S}; \mathfrak{c})$ in the second requirement can be replaced with $\text{MNX}(\Gamma'; \mathfrak{c})$. This is a consequence of Theorems 5.4.2 and 5.5.2.

As per Observation ii there will always be a dynamic trace, but of course the dynamic trace consisting of the whole game itself is of no use in reducing the effort needed to analyze the game. The next section deals with the discovery of smaller dynamic traces.

Games commonly have more than one dynamic trace. The dynamic trace property is hereditary in some sense; namely, any supergame of a dynamic trace is itself also a dynamic trace.

Theorem 9.1.2. Let $\Gamma = \langle X^{\mathcal{S}}, f \rangle$ and $\mathfrak{c} \in \mathfrak{C}$, and let $\Gamma' = \langle X^{\mathcal{S}'}, f' \rangle$ be a dynamic trace in $(\Gamma, \mathfrak{c})$. Then for any $\mathcal{S}''$ with $\mathcal{S}' \subseteq \mathcal{S}'' \subseteq \mathcal{S}$ the game $\Gamma' * \mathcal{S}''$ is also a dynamic trace in $(\Gamma, \mathfrak{c})$.

There need not be a unique smallest dynamic trace in a given game. A game can have several incomparable dynamic traces. A trivial example is a game with outcome function $\xi \mapsto \xi_0 \vee \xi_1 \vee \xi_2 \vee \ldots \vee \xi_{k-1}$, where $\langle X^{\{i,j\}}, \xi \mapsto \xi_i \vee \xi_j \rangle$ is a dynamic trace for MAX for any $0 \leqslant i < j < k$.

Another way of creating bigger dynamic traces from smaller ones is combining two or more dynamic traces.

Theorem 9.1.3. Let $\Gamma = \langle X^{\mathcal{S}}, f \rangle$ and $\mathfrak{c} \in \mathfrak{C}$. Let Γ'_0 and Γ'_1 be dynamic traces in $(\Gamma, \mathfrak{c})$. If they are dynamic traces for MAX, then $\Gamma'_0 \vee \Gamma'_1$ is a dynamic trace in $(\Gamma, \mathfrak{c})$. If they are dynamic traces for MIN, then $\Gamma'_0 \wedge \Gamma'_1$ is a dynamic trace in $(\Gamma, \mathfrak{c})$.

> *Corollary i:* If $\Gamma'_0, \Gamma'_1, \ldots, \Gamma'_{k-1}$ are dynamic traces for MAX in $(\Gamma, \mathfrak{c})$, then $\bigvee_i \Gamma'_i$ is a dynamic trace in $(\Gamma, \mathfrak{c})$. If they are dynamic traces for MIN in $(\Gamma, \mathfrak{c})$ then $\bigwedge_i \Gamma'_i$ is a dynamic trace for MIN in $(\Gamma, \mathfrak{c})$.

9.2 Mustplay

Closely related to dynamic traces are what Hayward *et al* called *mustplays*. Where a dynamic trace is a smaller game that guarantees a win by using just the dynamic trace, a mustplay is a smaller game that guarantees a loss when it is *not* used.

Definition 9.2.1. Let $\Gamma = \langle X^{\mathcal{S}}, f \rangle$ and $\Gamma' = \langle X^{\mathcal{S}'}, f' \rangle$ for some $\mathcal{S}' \subseteq \mathcal{S}$. Then Γ' is **a mustplay** in (Γ, MAX) if the following two conditions are satisfied:

- $\Gamma' \geqslant \Gamma$;
- $\text{MNX}(\Gamma' * \mathcal{S}*; \text{MIN}) = -1$.

Similarly, Γ' is a mustplay in (Γ, MIN) if $\Gamma' \leqslant \Gamma$ and $\text{MNX}(\Gamma' * \mathcal{S}*; \text{MAX}) = +1$.

> *Observation i:* If Γ' is a mustplay in $(\Gamma, \mathfrak{c})$ then Γ' is a dynamic trace in $(\Gamma, \mathfrak{c}) \oplus \mathfrak{m}$ for any move $\mathfrak{m} \in \mathfrak{M}(\mathcal{S} \setminus \mathcal{S}')$.

The definition is quite similar to the definition for dynamic traces, and Observation 9.2.1:i establishes a strong link. The implication of Observation 9.2.1:i follows because $\Gamma' = \Gamma'/\mathfrak{m}$ when $\mathfrak{m} \notin \mathcal{S}'$, which makes the requirements for Definitions 9.1.1 and 9.2.1 identical.

The 9.2.1:i implication does not work in the other direction: If some move $\mathfrak{m}$ produces a dynamic trace Γ' for the opponent, then Γ' might still not be a mustplay. This happens when $\mathfrak{m}$ was an unfortunate move that actually benefitted the opponent. In the next section the conditions will be outlined in which mustplays can be derived from dynamic traces.

As the term suggests, the purpose of identifying mustplays is the following theorem:

Theorem 9.2.2. Let $\Gamma = \langle X^{\mathcal{S}}, f \rangle$, and let $\Gamma' = \langle X^{\mathcal{S}'}, f' \rangle$ be a mustplay in $(\Gamma, \mathfrak{c})$. Let $\mathfrak{m} = \chi^v \in$

$\mathfrak{M}(\mathcal{S})$. If $v \notin \mathcal{S}'$ then $\mathfrak{m}$ is a losing move in $(\Gamma, \mathfrak{c})$.

Theorem 9.1.2 has an obvious analogue for mustplays. If $\Gamma' = \langle X^{\mathcal{S}'}, f' \rangle$ is a mustplay for MAX, and $\Gamma'' = \langle X^{\mathcal{S}''}, f'' \rangle \geqslant \Gamma'$ with $\mathcal{S}'' \subseteq \mathcal{S}'$, then Γ'' is also a mustplay for MAX.

As dynamic traces combine to form a bigger dynamic trace, so do mustplays combine to form a smaller mustplay. If $\Gamma'_0 = \langle X^{\mathcal{S}'_0}, f'_0 \rangle$ and $\Gamma'_1 = \langle X^{\mathcal{S}'_1}, f'_1 \rangle$ are mustplays for MAX, then the previous observation shows that some game $\Gamma'' = \langle X^{\mathcal{S}'_0 \cap \mathcal{S}'_1}, f'' \rangle$ satisfying $\Gamma'' \geqslant \Gamma'_0$ and $\Gamma'' \geqslant \Gamma'_1$ is also a mustplay for MAX. Such a game does exist, because at least the trivial game $\langle X^{\mathcal{S}'_0 \cap \mathcal{S}'_1}, + \rangle$ satisfies the requirement.

9.3 Recursive Detection of Dynamic Traces

Fortunately, dynamic traces and mustplays smaller than the game itself often exist, and they can be discovered dynamically through mutual recursion without using any game-specific knowledge. They can be used as a safe pruning mechanism in game tree search, guaranteed to prune at least as many branches as $\alpha - \beta$ search[1] and only to prune provably irrelevant branches.

The recursion starts with the following, literally trivial, theorem.

Theorem 9.3.1. If Γ is trivial, with $\Gamma = \langle X^{\mathcal{S}}, t \rangle$, then $\langle \varnothing, t \rangle$ is a dynamic trace for $\lambda^{-1}(t)$ in $(\Gamma, \mathfrak{c})$.

When a winning move is found, a dynamic trace can be constructed based on a dynamic trace of the resulting subgame by adding the move to the dynamic trace:

Theorem 9.3.2. Let $\Gamma = \langle X^{\mathcal{S}}, f \rangle$ and $\mathfrak{m} = \chi^v \in \mathfrak{M}(\mathcal{S})$. If $\mathfrak{m}$ is a winning move in $(\Gamma, \mathfrak{c})$, and $\Gamma' = \langle X^{\mathcal{S}'}, f' \rangle$ is a dynamic trace for $\mathfrak{c}$ in $(\Gamma, \mathfrak{c}) \oplus \mathfrak{m}$, then the following game is a dynamic trace for $\mathfrak{c}$ in $(\Gamma, \mathfrak{c})$:

$$\Gamma'' = \left\langle X^{\mathcal{S}'+v}, \xi \mapsto f'(\xi) \wedge (\xi_v = \chi) \right\rangle.$$

This type of construction was seen earlier in Theorem 6.1.4, ensuring that $\Gamma''/\mathfrak{m} = \Gamma'$.

When a position is a loss, the regular minimax formula needs to examine all possible moves to confirm the loss. Dynamic traces improve this by proving losses without examining all the moves. This is possible based on the observation that when all moves to one particular element are found to be losses, then a mustplay Γ' is established which means that all moves outside of Γ' can be discarded without examining them.

Theorem 9.3.3. Let $\Gamma = \langle X^{\mathcal{S}}, f \rangle$ and $v \in \mathcal{S}$. For each $\chi \in X$ let Γ'_χ be a dynamic trace in $(\Gamma, \mathfrak{c}) \oplus \chi^v$. If all moves $\{\chi^v\}_{\chi \in X}$ are losses in $(\Gamma, \mathfrak{c})$, then $\bigwedge_{\chi \in X} \Gamma'_\chi$ is a mustplay in $(\Gamma, \mathfrak{c})$ if $\mathfrak{c} = $ MAX, and $\bigvee_{\chi \in X} \Gamma'_\chi$ is a mustplay in $(\Gamma, \mathfrak{c})$ if $\mathfrak{c} = $ MIN.

[1]The standard game tree search algorithm; see Section 11.1.

Theorem 9.3.3 examines all possible moves to one specific element v, and if they all lose then potentially several other moves are proved to be losses as well. If v admits a rational move, then it may be expected that it is sufficient to examine only one rational move. This is indeed the case, as per the following theorem.

Theorem 9.3.4. Let $\Gamma = \langle X^{\mathcal{S}}, f \rangle$ and $v \in \mathcal{S}$. If for some rational move $\mathfrak{m} \in \mathfrak{M}(\mathcal{S})$ we have that $\mathfrak{m}$ is a losing move in $(\Gamma, \mathfrak{c})$ and $\Gamma' = \langle X^{\mathcal{S}'}, f' \rangle$ is a dynamic trace in $(\Gamma, \mathfrak{c}) \oplus \mathfrak{m}$, then Γ' is a mustplay in $(\Gamma, \mathfrak{c})$.

By using the previous two theorems, examining losing moves identifies mustplays. When a number of mustplays are identified that have no common intersection, there apparently are no moves that can counter all the threats. The position has then been proved to be a loss. Moreover, the collection of mustplays combine to form a dynamic trace.

Theorem 9.3.5. Let $\Gamma = \langle X^{\mathcal{S}}, f \rangle$, and let $\{\Gamma'_i = \langle X^{\mathcal{S}'_i}, f'_i \rangle\}_{i \in \mathbb{Z}_k}$ be mustplays in $(\Gamma, \mathfrak{c})$, with $\bigcap_{i \in \mathbb{Z}_k} \mathcal{S}'_i = \varnothing$. Then $\text{NGX}(\Gamma; \mathfrak{c}) = -1$. Moreover, if $\mathfrak{c} = \text{MAX}$ then $\bigwedge_{i \in \mathbb{Z}_k} \Gamma'_i$ is a dynamic trace in (Γ, MAX), and if $\mathfrak{c} = \text{MIN}$ then $\bigvee_{i \in \mathbb{Z}_k} \Gamma'_i$ is a dynamic trace in (Γ, MIN).

If a position is a loss, then the existence of a collection of mustplays satisfying the requirement of Theorem 9.3.5 is guaranteed. This is trivially true since each $v \in \mathcal{S}$ leads to a mustplay that does not contain v, according to the constructions of Theorems 9.3.3 and 9.3.4.

In practice it will be beneficial to keep the dynamic traces and mustplays as small as possible. The smaller a dynamic trace or mustplay, the more moves are discarded without needing to be examined.

9.4 Dynamic Trace Patterns

Both dynamic traces and mustplays are defined as set colouring games in their own right. However, the practical use of dynamic traces and mustplays lies in discarding moves that are proven losses without having been examined. For this purpose it is actually sufficient to keep track of only the *colour spaces* of the games in question.

Definition 9.4.1. Let $\Gamma = \langle X^{\mathcal{S}}, f \rangle$, $\Gamma' = \langle X^{\mathcal{S}'}, f' \rangle$, and $\mathfrak{c} = \mathfrak{C}$. If Γ' is a dynamic trace in $(\Gamma, \mathfrak{c})$ then $\mathcal{S}'$ is **a dynamic trace pattern** in $(\Gamma, \mathfrak{c})$. If Γ' is a mustplay in $(\Gamma, \mathfrak{c})$ then $\mathcal{S}'$ is **a mustplay pattern** in $(\Gamma, \mathfrak{c})$.

The theorems of Section 9.3 then give the following rules. Let $\Gamma = \langle X^{\mathcal{S}}, f \rangle$, then:

- If Γ is trivial, with $\Gamma = \langle X^{\mathcal{S}}, t \rangle$, then $\varnothing$ is a dynamic trace pattern for $\lambda^{-1}(t)$ in $(\Gamma, \mathfrak{c})$.
- Let $\mathfrak{m} = \chi^v \in \mathfrak{M}(\mathcal{S})$. If $\mathfrak{m}$ is a winning move in $(\Gamma, \mathfrak{c})$, and $\mathcal{S}'$ is a dynamic trace pattern in $(\Gamma, \mathfrak{c}) \oplus \mathfrak{m}$, then $\mathcal{S}' + v$ is a dynamic trace pattern for $\mathfrak{c}$ in $(\Gamma, \mathfrak{c})$.
- Let $v \in \mathcal{S}$. For each $\chi \in X$ let $\mathcal{S}'_\chi$ be a dynamic trace pattern in $(\Gamma, \mathfrak{c}) \oplus \chi^v$. If all moves $\{\chi^v\}_{\chi \in X}$ are losses in $(\Gamma, \mathfrak{c})$, then $\bigcup_{\chi \in X} \mathcal{S}'_\chi$ is a mustplay pattern in $(\Gamma, \mathfrak{c})$.

- If for some rational move $\mathfrak{m} \in \mathfrak{M}(\mathcal{S})$ we have that $\mathfrak{m}$ is a losing move in $(\Gamma, \mathfrak{c})$ and $\mathcal{S}'$ is a dynamic trace pattern in $(\Gamma, \mathfrak{c}) \oplus \mathfrak{m}$, then $\mathcal{S}'$ is a mustplay pattern in $(\Gamma, \mathfrak{c})$.
- Let $\{\mathcal{S}'_i\}_{i \in \mathbb{Z}_k}$ be mustplay patterns in $(\Gamma, \mathfrak{c})$, with $\bigcap_{i \in \mathbb{Z}_k} \mathcal{S}'_i = \varnothing$. Then $\bigcup_{i \in \mathbb{Z}_k} \mathcal{S}'_i$ is a dynamic trace pattern for $\bar{\mathfrak{c}}$ in $(\Gamma, \mathfrak{c})$.

A search algorithm based on these observations will be presented in Section 11.2.

9.5 Proofs

Theorem 9.1.2. Let $\Gamma = \langle X^{\mathcal{S}}, f \rangle$ and $\mathfrak{c} \in \mathfrak{C}$, and let $\Gamma' = \langle X^{\mathcal{S}'}, f' \rangle$ be a dynamic trace in $(\Gamma, \mathfrak{c})$. Then for any $\mathcal{S}''$ with $\mathcal{S}' \subseteq \mathcal{S}'' \subseteq \mathcal{S}$ the game $\Gamma' * \mathcal{S}''$ is also a dynamic trace in $(\Gamma, \mathfrak{c})$.

Proof. Note that $(\Gamma' * \mathcal{S}'') * \mathcal{S} = \Gamma' * \mathcal{S}$ since $\mathcal{S}'' \subseteq \mathcal{S}$. Consider first the case where Γ' is a dynamic trace for MAX, so that $\Gamma' \leqslant \Gamma$, which by definition means $\Gamma' * \mathcal{S} \leqslant \Gamma$. We have $\Gamma' * \mathcal{S}'' \leqslant \Gamma$ since $(\Gamma' * \mathcal{S}'') * \mathcal{S} = \Gamma' * \mathcal{S} \leqslant \Gamma$. Also, $\text{MNX}((\Gamma' * \mathcal{S}'') * \mathcal{S}; \mathfrak{c}) = \text{MNX}(\Gamma' * \mathcal{S}; \mathfrak{c}) = +1$, fulfilling both conditions. The case where Γ' is a dynamic trace for MIN is entirely analogous. □

Theorem 9.1.3. Let $\Gamma = \langle X^{\mathcal{S}}, f \rangle$ and $\mathfrak{c} \in \mathfrak{C}$. Let Γ'_0 and Γ'_1 be dynamic traces in $(\Gamma, \mathfrak{c})$. If they are dynamic traces for MAX, then $\Gamma'_0 \vee \Gamma'_1$ is a dynamic trace in $(\Gamma, \mathfrak{c})$. If they are dynamic traces for MIN, then $\Gamma'_0 \wedge \Gamma'_1$ is a dynamic trace in $(\Gamma, \mathfrak{c})$.

> *Corollary i:* If $\Gamma'_0, \Gamma'_1, \ldots, \Gamma'_{k-1}$ are dynamic traces for MAX in $(\Gamma, \mathfrak{c})$, then $\bigvee_i \Gamma'_i$ is a dynamic trace in $(\Gamma, \mathfrak{c})$. If they are dynamic traces for MIN in $(\Gamma, \mathfrak{c})$ then $\bigwedge_i \Gamma'_i$ is a dynamic trace for MIN in $(\Gamma, \mathfrak{c})$.

Proof. Theorem 6.4.1 and its corollaries immediately imply the desired properties. □

Theorem 9.2.2. Let $\Gamma = \langle X^{\mathcal{S}}, f \rangle$, and let $\Gamma' = \langle X^{\mathcal{S}'}, f' \rangle$ be a mustplay in $(\Gamma, \mathfrak{c})$. Let $\mathfrak{m} = \chi^v \in \mathfrak{M}(\mathcal{S})$. If $v \notin \mathcal{S}'$ then $\mathfrak{m}$ is a losing move in $(\Gamma, \mathfrak{c})$.

Proof. First consider the case $\mathfrak{c} = \text{MAX}$. According to Observation 9.2.1:i, Γ' is a dynamic trace in $(\Gamma, \mathfrak{c}) \oplus \mathfrak{m} = (\Gamma/\mathfrak{m}, \text{MIN})$. Since Γ' is a mustplay in (Γ, MAX) we have $-1 = \text{MNX}(\Gamma' * \mathcal{S}*; \text{MIN}) = \text{MNX}(\Gamma' * (\mathcal{S} - v); \text{MIN})$ according to Theorem 5.4.2 since $\mathcal{S}*$ and $\mathcal{S} - v$ have the same parity. Therefore, in order to meet the definition of dynamic traces, the owner of the dynamic trace Γ' in $(\Gamma, \mathfrak{c}) \oplus \mathfrak{m})$ must be MIN. Observation 9.1.1:i then implies that $\text{MNX}(\Gamma/\mathfrak{m}; \text{MIN}) = -1$, which means that $\mathfrak{m}$ was a losing move in (Γ, MAX). The proof for the case $\mathfrak{c} = \text{MIN}$ is analogous. □

Theorem 9.3.1. If Γ is trivial, with $\Gamma = \langle X^{\mathcal{S}}, t \rangle$, then $\langle \varnothing, t \rangle$ is a dynamic trace for $\lambda^{-1}(t)$ in $(\Gamma, \mathfrak{c})$.

Proof. Applying the definitions shows that $\langle \varnothing, t \rangle \geqslant \Gamma$ and $\text{MNX}(\langle \varnothing, t \rangle * \mathcal{S}; \mathfrak{c}) = +1$ when $t = +1$, and when $t = -1$ the requirements are met similarly. □

For Theorem 9.3.2 first a lemma about combining supergames and subgames:

Lemma 9.5.1. Let $\Gamma = \langle X^{\mathcal{S}}, f\rangle$ with $\mathcal{S} \subseteq \mathcal{S}^*$, and $\psi \in X^{\mathcal{S}^*}$. Then $\Gamma * \mathcal{S}^*/\psi = (\Gamma/\psi) * (\mathcal{S}^* \setminus \mathcal{A}(\psi))$.

Proof. Writing out the expressions reveals that both sides of the equation specify the following game:

$$\left\langle X^{\mathcal{S}^*\setminus\mathcal{A}(\psi)}, \xi \mapsto f\big((\xi\overline{\psi}) \searrow \mathcal{S}\big)\right\rangle. \qquad \square$$

Theorem 9.3.2. Let $\Gamma = \langle X^{\mathcal{S}}, f\rangle$ and $\mathfrak{m} = \chi^v \in \mathfrak{M}(\mathcal{S})$. If $\mathfrak{m}$ is a winning move in $(\Gamma, \mathfrak{c})$, and $\Gamma' = \langle X^{\mathcal{S}'}, f'\rangle$ is a dynamic trace for $\mathfrak{c}$ in $(\Gamma, \mathfrak{c}) \oplus \mathfrak{m}$, then the following game is a dynamic trace for $\mathfrak{c}$ in $(\Gamma, \mathfrak{c})$:

$$\Gamma'' = \left\langle X^{\mathcal{S}'+v}, \xi \mapsto f'(\xi) \wedge (\xi_v = \chi)\right\rangle.$$

Proof. This type of construction was seen earlier in Theorem 6.1.4, ensuring that $\Gamma''/\mathfrak{m} = \Gamma'$. Recall that $(\Gamma, \mathfrak{c}) \oplus \mathfrak{m} = (\Gamma/\mathfrak{m}, \bar{\mathfrak{c}})$, and that $\mathcal{S}' = \mathcal{S} - v$. Consider the case where $\mathfrak{c} = \text{MAX}$. This guarantees that $\text{MNX}(\Gamma' * (\mathcal{S} - v); \bar{\mathfrak{c}}) = +1$, and $\Gamma' \leqslant \Gamma/\mathfrak{m}$, meaning $f'(\psi^*) \leqslant f(\psi^*\mathfrak{m})$ for any $\psi^* \in \check{X}^{\mathcal{S}}$. The two requirements are met as follows.

- Let f'' be the outcome function specified for Γ'', and let $\mathcal{S}'' = \mathcal{S}' + v$. Let $\psi^* \in \check{X}^{\mathcal{S}}$. If $f''(\psi^*) = +1$ then this means $f'(\psi^*) = f'(\psi^* \searrow \mathcal{S}'' \searrow \mathcal{S}') = f'(\psi^* \searrow \mathcal{S}') = f'(\psi^*) = +1$ and $\psi^*_v = \chi$. The latter implies that $\psi^*\mathfrak{m} = \psi^*$. With the former we then have $f(\psi^*) = f(\psi^*\mathfrak{m}) \geqslant f'(\psi^*\mathfrak{m}) = f'(\psi^*) = +1$, where $\psi^*\mathfrak{m} \searrow= \psi^* \searrow \mathcal{S}'$ since $\mathcal{D}(\mathfrak{m}) \notin \mathcal{S}'$. Apparently $f''(\psi^*) = +1$ implies $f(\psi^*) = +1$, and therefore $\Gamma'' \leqslant \Gamma$.
- With Lemma 9.5.1 implying $\Gamma' * (\mathcal{S} \setminus v) = (\Gamma''/\mathfrak{m}) * (\mathcal{S} \setminus v) = (\Gamma'' * \mathcal{S})/\mathfrak{m}$ we have

$$\text{MNX}(\Gamma'' * \mathcal{S}; \text{MAX}) \geqslant \text{MNX}((\Gamma'' * \mathcal{S})/\mathfrak{m}; \text{MIN}) = \text{MNX}(\Gamma' * (\mathcal{S} \setminus v); \text{MIN}) = +1.$$

The proof for $\mathfrak{c} = \text{MIN}$ is analogous. $\square$

In order to prove the next theorems, first a lemma about comparing a game with subgames of another game.

Lemma 9.5.2. Let $\Gamma = \langle X^{\mathcal{S}}, f\rangle$ and $\Gamma' = \langle X^{\mathcal{S}'}, f'\rangle$, and let $v \in \mathcal{S}$. If $\Gamma' \leqslant \Gamma/\chi^v$ for all $\chi \in X$, then $\Gamma' \leqslant \Gamma$. Similarly, $\forall_{\chi \in X}[\Gamma' \geqslant \Gamma/\chi^v] \Longrightarrow \Gamma' \geqslant \Gamma$.

> *Corollary i:* Let $\mathcal{S}'' \subseteq \mathcal{S}$. If $\Gamma' \leqslant \Gamma/\psi$ for all $\psi \in \check{X}^{\mathcal{S}''}$, then $\Gamma' \leqslant \Gamma$. Similarly, $\forall_{\psi \in \check{X}^{\mathcal{S}''}}[\Gamma' \geqslant \Gamma/\psi] \Longrightarrow \Gamma' \geqslant \Gamma$.
>
> *Corollary ii:* If $\Gamma' \leqslant \Gamma/\mathfrak{m}$ for some move $\mathfrak{m}$ that is rational for MIN, then $\Gamma' \leqslant \Gamma$. If $\Gamma' \geqslant \Gamma/\mathfrak{m}$ for some move $\mathfrak{m}$ that is rational for MAX, then $\Gamma' \leqslant \Gamma$.

Proof. Let $\mathcal{S}^* = \mathcal{S} \cup \mathcal{S}'$, and let $\psi^* \in \mathcal{S}^*$. Put $\chi = \psi^*_v$, so that $\psi^*\chi^v = \psi^*$. If $\Gamma' \leqslant \Gamma/\chi^v$ then $f'(\psi^* \searrow \mathcal{S}') \leqslant f\big((\psi^* \searrow \mathcal{S})\chi^v\big) = f(\psi^*\chi^v \searrow \mathcal{S}) = f(\psi^* \searrow \mathcal{S})$ using Lemma 2.2.2:i. This satisfies the requirement for $\Gamma' \leqslant \Gamma$. If $\Gamma' \geqslant \Gamma/\chi^v$ then similarly $f'(\psi^* \searrow \mathcal{S}') \geqslant f(\psi^* \searrow \mathcal{S})$. Corollary i follows by induction to $|\mathcal{S}''|$, and Corollary ii follows from the fact that $\Gamma/\mathfrak{m} \leqslant \Gamma/\mathfrak{m}'$ if $\mathfrak{m}$ is rational for MIN, and $\Gamma/\mathfrak{m} \geqslant \Gamma/\mathfrak{m}'$ if $\mathfrak{m}$ is rational for MAX. $\square$

Theorem 9.3.3. Let $\Gamma = \langle X^{\mathcal{S}}, f\rangle$ and $v \in \mathcal{S}$. For each $\chi \in X$ let Γ'_χ be a dynamic trace in $(\Gamma, \mathfrak{c}) \oplus \chi^v$. If all moves $\{\chi^v\}_{\chi \in X}$ are losses in $(\Gamma, \mathfrak{c})$, then $\bigwedge_{\chi\in X} \Gamma'_\chi$ is a mustplay in $(\Gamma, \mathfrak{c})$ if $\mathfrak{c} = \text{MAX}$, and $\bigvee_{\chi \in X} \Gamma'_\chi$ is a mustplay in $(\Gamma, \mathfrak{c})$ if $\mathfrak{c} = \text{MIN}$.

Proof. Consider the case where $\mathfrak{c} = \text{MAX}$, then the Γ'_χ are dynamic traces for MIN. Put $\Gamma' = \langle X^{\mathcal{S}'}, f'\rangle = \bigwedge_\chi \Gamma'_\chi$, so that $\mathcal{S}' = \bigcup_\chi \mathcal{S}_\chi$ and $v \notin \mathcal{S}'$. The latter is because v does not occur in the colour space of $(\Gamma, \mathfrak{c}) \oplus \chi^v$ for any $\chi \in X$. According to Theorem 9.1.3, Γ' is a dynamic trace for MIN in $(\Gamma, \text{MAX}) \oplus \chi^v$ for every $\chi \in X$. Now let $\mathfrak{m} \in \mathfrak{M}(\mathcal{S} \setminus \bigcup_\chi \mathcal{S}_\chi)$. The game Γ' has the following two properties:

- For every $\chi \in X$ we have $\Gamma' \geqslant \Gamma/\chi^v$ because Γ' is a dynamic trace for MIN in $(\Gamma, \text{MAX}) \oplus \chi^v$. By Lemma 9.5.2 this implies $\Gamma' \geqslant \Gamma$. Since $\mathcal{D}(\mathfrak{m}) \notin \mathcal{S}'$ we have $\Gamma'/\mathfrak{m} = \Gamma' \geqslant \Gamma$.
- Note that $\text{MNX}(\Gamma' * (\mathcal{S} - v); \text{MIN}) = -1$ because Γ' is a dynamic trace for MIN in Γ/χ^v. The sets $\mathcal{S} - v$ and $\mathcal{S} - w$ have equal parity and both contain $\mathcal{S}'$. Therefore $\text{MNX}(\Gamma' * (\mathcal{S} - w); \text{MIN}) = \text{MNX}(\Gamma' * (\mathcal{S} - v); \text{MIN}) = -1$.

These two properties establish that Γ' is a dynamic trace for MIN in Γ/χ^w, which means that χ^w was a losing move in (Γ, MAX). The proof for the case $\mathfrak{c} = \text{MIN}$ is analogous. □

Theorem 9.3.4. Let $\Gamma = \langle X^{\mathcal{S}}, f\rangle$ and $v \in \mathcal{S}$. If for some rational move $\mathfrak{m} \in \mathfrak{M}(\mathcal{S})$ we have that $\mathfrak{m}$ is a losing move in $(\Gamma, \mathfrak{c})$ and $\Gamma' = \langle X^{\mathcal{S}'}, f'\rangle$ is a dynamic trace in $(\Gamma, \mathfrak{c}) \oplus \mathfrak{m}$, then Γ' is a mustplay in $(\Gamma, \mathfrak{c})$.

Proof. Put $v = \mathcal{D}(\mathfrak{m})$. Consider the case where $\mathfrak{c} = \text{MAX}$. Let $\chi \in X$. Since $\mathfrak{m}$ is rational for MAX we have $\Gamma/\mathfrak{m} \geqslant \Gamma/\chi^v$ and $\text{MNX}(\Gamma/\mathfrak{m} * (\mathcal{S} - v); \text{MIN}) \geqslant \text{MNX}(\Gamma/\chi^v * (\mathcal{S} - v); \text{MIN})$. Since Γ' is a dynamic trace for MIN in $(\Gamma/\mathfrak{m}, \text{MIN})$ we have $\Gamma' \geqslant \Gamma/\mathfrak{m}$ and $\text{MNX}(\Gamma/\mathfrak{m} * (\mathcal{S} - v); \text{MIN}) = -1$. Combining this gives $\Gamma' \geqslant \Gamma/\chi^v$ and $\text{MNX}(\Gamma/\chi^v * (\mathcal{S} - v); \text{MIN}) = -1$, so Γ' is a dynamic trace for MIN in Γ/χ^v. Since this holds for any $\chi \in X$, Theorem 9.3.3 applies. The proof for $\mathfrak{c} = \text{MIN}$ is analogous. □

Theorem 9.3.5. Let $\Gamma = \langle X^{\mathcal{S}}, f\rangle$, and let $\{\Gamma'_i = \langle X^{\mathcal{S}'_i}, f'_i\rangle\}_{i \in \mathbb{Z}_k}$ be mustplays in $(\Gamma, \mathfrak{c})$, with $\bigcap_{i\in\mathbb{Z}_k} \mathcal{S}'_i = \varnothing$. Then $\text{NGX}(\Gamma; \mathfrak{c}) = -1$. Moreover, if $\mathfrak{c} = \text{MAX}$ then $\bigwedge_{i\in\mathbb{Z}_k} \Gamma'_i$ is a dynamic trace in (Γ, MAX), and if $\mathfrak{c} = \text{MIN}$ then $\bigvee_{i \in \mathbb{Z}_k} \Gamma'_i$ is a dynamic trace in (Γ, MIN).

Proof. Let $\mathfrak{m} \in \mathfrak{M}(\mathcal{S})$, then since $\bigcap_{i\in\mathbb{Z}_k} \mathcal{S}'_i = \varnothing$ there exists a Γ'_i such that $\mathfrak{m} \notin \mathfrak{M}(\mathcal{S}'_i)$. According to Theorem 9.2.2, $\mathfrak{m}$ is a losing move in $(\Gamma, \mathfrak{c})$. Since all moves apparently lose, we have $\text{NGX}(\Gamma; \mathfrak{c}) = -1$. For the dynamic trace proof.

Consider first the case where $\mathfrak{c} = \text{MAX}$. Put $\Gamma' = \bigwedge_{i\in\mathbb{Z}_k} \Gamma'_i$.

- For each Γ_i we have $\Gamma_i \geqslant \Gamma$, which by Theorem 6.3.6 implies $\Gamma' = \bigwedge_{i\in\mathbb{Z}_k} \Gamma'_i \geqslant \Gamma$.
- Let $\mathfrak{m} = \chi^v \in \mathfrak{M}(\mathcal{S})$, then

$$(\Gamma' * \mathcal{S}, \text{MAX}) \oplus \mathfrak{m} = (\Gamma'/\mathfrak{m} * (\mathcal{S} - v), \text{MIN}) = \Big(\big(\bigwedge_{i\in\mathbb{Z}_k} \Gamma'_i/\mathfrak{m}\big) * (\mathcal{S} - v), \text{MIN}\Big)$$

and we have $\text{MNX}\big((\bigwedge_{i\in\mathbb{Z}_k} \Gamma'_i/\mathfrak{m}) * (\mathcal{S} - v); \text{MIN}\big) = \text{MNX}\big((\bigwedge_{i\in\mathbb{Z}_k} \Gamma'_i/\mathfrak{m}) * \mathcal{S}*; \text{MIN}\big)$ by Theorem 5.4.2 because $\mathcal{S} - v$ and $\mathcal{S}*$ have the same parity. Since $\bigcap_{i \in \mathbb{Z}_k} \mathcal{S}'_i = \varnothing$ there must be a $j \in \mathbb{Z}_k$ with $v \notin \mathcal{S}'_j$. For Γ'_j we then have $\Gamma'_j/\mathfrak{m} = \Gamma'_j$ and therefore $\textsc{mnx}\big((\Gamma'_j/\mathfrak{m}) * \mathcal{S}*; \textsc{min}\big) = \textsc{mnx}(\Gamma'_j * \mathcal{S}*; \textsc{min})$. Since $\Gamma' \leqslant \Gamma_j$ by Observation 6.3.2:iii, this gives

$$\textsc{mnx}\Big(\big(\bigwedge_{i \in \mathbb{Z}_k} \Gamma'_i/\mathfrak{m}\big) * \mathcal{S}*; \textsc{min}\Big) \leqslant \textsc{mnx}\big((\Gamma'_j/\mathfrak{m}) * \mathcal{S}*; \textsc{min}\big) = \textsc{mnx}(\Gamma'_j * \mathcal{S}*; \textsc{min}) = -1.$$

Therefore $\bigwedge_{i \in \mathbb{Z}_k} \Gamma'_i$ is a dynamic trace for MIN in $(\Gamma, \mathfrak{c})$. Also, from Observation 9.1.1:i this means $\textsc{mnx}(\Gamma; \textsc{max}) = -1$. Analogously, for the case $\mathfrak{c} = \textsc{min}$ we obtain that $\bigvee_{i \in \mathbb{Z}_k} \Gamma'_i$ is a dynamic trace for MAX in $(\Gamma, \mathfrak{c})$, and therefore $\textsc{mnx}(\Gamma; \textsc{min}) = +1$. In either case we have $\textsc{ngx}(\Gamma; \mathfrak{c}) = -1$. □

Part II

Hex and Computation

Chapter 10

Properties of Hex

Since Hex and, more generally, the Shannon game are played on graphs, they exhibit more structure than set colouring games in general. In particular, the graph makes it possible to consider localized properties that occur near a certain element. The purpose of this chapter is to review all known theoretical properties of Hex and use the theory established in Part I to prove them.

10.1 No draws

The most fundamental observations about Hex are that the game can never end in a draw, and that there must exist a winning strategy for the player who moves first. The "no draw" property is inherent to set colouring games, as well as the Shannon game. However, Hex is traditionally presented with each player having a certain connection goal, rather than one player trying to connect and the other player trying to block. It is a particular property of the Hex graph that blocking a connection is equivalent to establishing a different connection.

Various proofs of the no-draw property have been given over the years [10, 14, 32]. The clearest proof is the one given by Gale, based on the fact that the board is planar and exactly three cells meet at every corner on the Hex board. Schensted called this the "mudcrack principle", and extended the game of Y and related games to be played on any mudcrack board. A board has the mudcrack property if the graph whose vertices correspond to the board cells and whose edges connect each pair of adjacent board cells is a planar triangulated graph. The mudcrack principle is also the basis for the no-draw proof for the game of Y using the reduction method [48].[1]

Figure 10.1 shows Gale's proof. Any planar mudcrack board where two contiguous strings of outer cells are chosen as one player's goal areas can be deformed continuously into a square as shown, in which the player in question attempts to connect the left and right sides. When the board

[1] See Section 4.6.

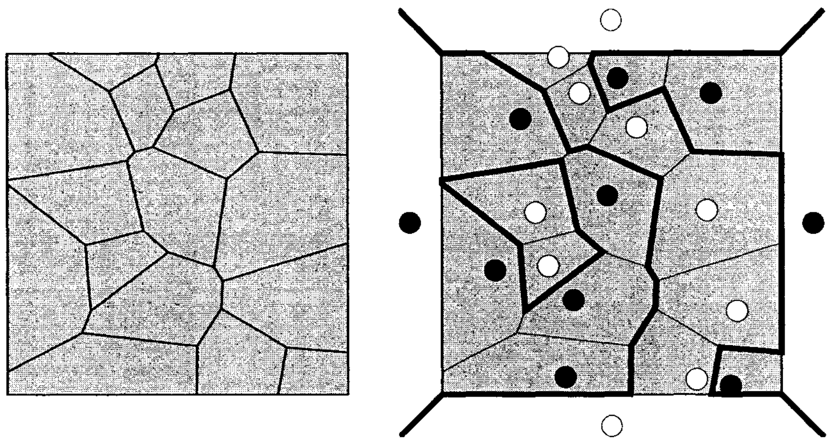


Figure 10.1: Gale's no-draw proof applied to a "mudcrack" board.

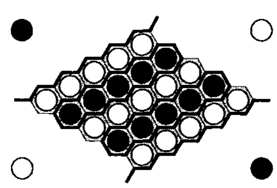


Figure 10.2: A difficulty in proving that *exactly* one player has a winning string.

is completely filled with pieces, all edges between oppositely coloured cells are highlighted. Since exactly three edges meet at any intersection, the highlighted edges define a subgraph of degree one and two, which must therefore necessarily consist of simple loops and paths. The paths must start and end at the only vertices of degree one, being at the four corners of the board. On either side of such a path is a string of pieces of one colour, connecting two opposite sides of the square, and thus creating a winning connection for one of the players. This implies that if the horizontal player has not connected the left and right sides, the vertical player must have connected the top and bottom.

To maintain the degree-3 property, the added corner edges must originate from the edge of a board cell, not from the intersection of two. It is therefore important that in "mudcrack Hex" the corner cells each belong to two edges, otherwise draws would be possible.

As Gale points out, this proof only shows that *at least* one of the players must have established a connection. The fact that the paths in Figure 10.1 cannot connect opposite corners of the board can be seen by orienting each highlighted edge such that there is a black cell on the left and a white cell on the right. But as the middle white string in Figure 10.2 shows, there can be winning strings whose boundary does not connect two corners. A topological proof that there cannot be winning chains of both colours can be outlined by imagining winning chains of opposing colours, extending the white chain to the leftmost and rightmost corner of the board via the border, extending the black chain to the top and bottom corner of the board, and then invoking a theorem to the effect that two interior diagonals in a quadrilateral must necessarily intersect.

10.2 First Player Win

The first player win property of Hex is usually proved using Nash's "strategy stealing" argument: If there were a winning strategy for the player to go second, then the first player could "steal" this strategy by making an irrelevant move, and then ignoring this move and applying the winning strategy. This strategy works because the extra piece can never be a disadvantage, a fact which itself would need to be proved.[2] The strategy also needs to cope with the situation where the move it recommends happens to be in the cell that already contains the irrelevant move; care then needs to be taken to show that the player can always make another irrelevant move.

Using the results of Chapter 3, the first player win property directly follows from Theorem 5.6.2. The anti-automorphism of the Hex board consists of reflecting the board in one of the diagonals and flipping the colour of every cell. The isotonicity of Hex follows from the fact that it is a coalition game, where the coalitions are the connecting paths.

10.3 Complexity

The Shannon game was the first commonly played game to be shown PSPACE-complete, by Even and Tarjan in 1976 [29]. Their construction uses a direct reduction from the QBF problem. Arratia proved that the game is still PSPACE-complete even if both players are restricted to colouring only nodes adjacent to Short's last move [8]. Despite the regular structure of the Hex board, Hex is no less complex than the Shannon game or QBF. Reisch proved in 1981 that Hex is PSPACE-complete as well [80].

The first step in Reisch's proof is to reduce QBF to "bipartite Geography". The game of Geography is played on a directed graph, where each player must move adjacent to the previous move. The first player unable to move loses. Next, Geography is reduced to "bipartite Geography on planar digraphs with degree $\leqslant 3$" by introducing gadgets that remove all edge crossings and all vertices with in-degree or out-degree larger than 3. The third step is to reduce this to the Shannon game played on undirected graphs with the same properties. Finally, those graphs are embedded in large Hex boards by setting up Hex positions that represent such graphs.

Where artificial intelligence approaches are concerned, what matters is not so much the asymptotic complexity, but the actual effort involved in playing the game on a fixed board size. The relevant measures are *game tree complexity* and *state space complexity*. Hex is compared with a variety of other games commonly played by humans and computers in [52]. Figure 10.3 graphs these complexities for various board sizes, as compared to some other well-known games that have been studied in artificial intelligence.

[2]See Theorem 5.5.3.

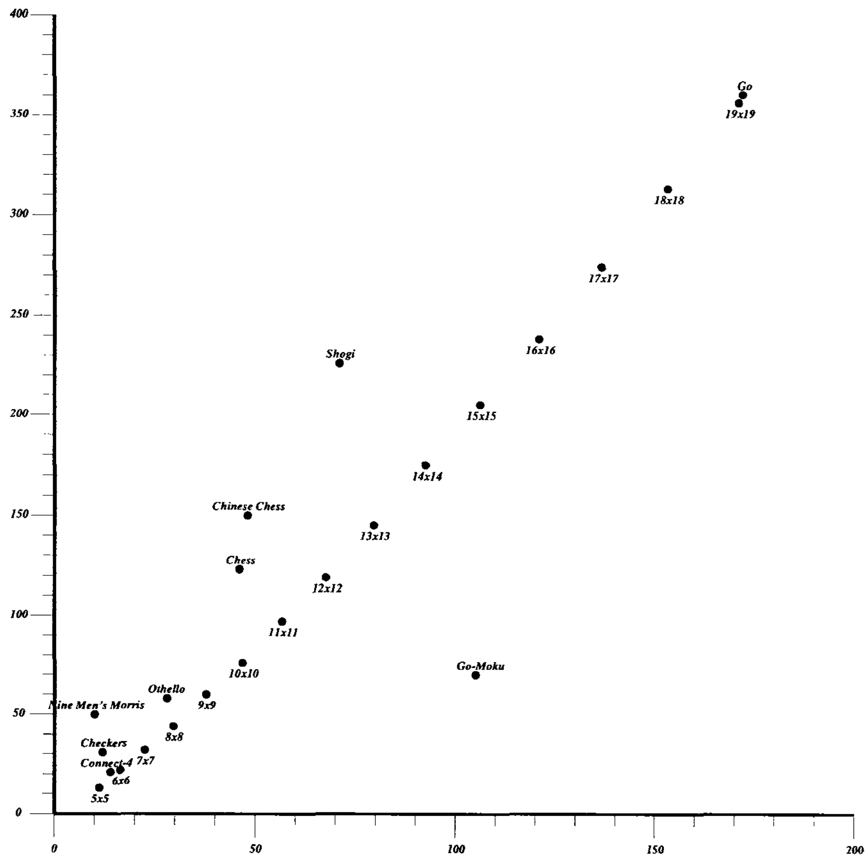


Figure 10.3: Logarithm of game-tree size (horizontal) and state-space size (vertical) for various games. Entries labelled "$n \times n$ refer to Hex on different board sizes.

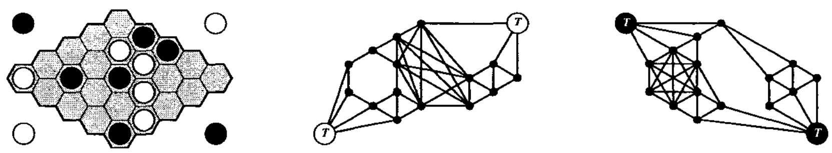


Figure 10.4: A Hex position and its two reduced graph representations as a Shannon game.

Both complexities can be readily calculated for Hex on a given board size. The state space complexity of Hex on a board of size $m \times n$ is almost equal to 3^{mn}, since each cell can be either empty or contain a black or white stone. The actual number is a bit less since the difference between the number of black and white stones cannot be more than one in a legal position. The game tree complexity would be $(mn)!$ if the game were played until the entire board is full. However, in practice the game ends long before the board is full. If the game typically ends when a fraction r of the cells filled, the game tree complexity would be an estimated $\frac{(mn)!}{((1-r)mn)!}$. From statistical surveys of actual Hex games it seems that $r \approx 0.4$ on average in games played between top human players.

10.4 Graph Representations

The empty Hex board can be represented as a Shannon game graph as discussed in Section 4.7. The game can then be played on this graph by colouring the vertices. In general, let $\Gamma = \langle \mathbb{T}^{\mathcal{S}}, f \rangle$ be a Shannon game with game graph $\mathcal{G}$, and let $v \in \mathcal{S}$. Suppose the move F^v is played, which in Shannon game terms means that v has been coloured with Cut's colour. If at the end of the game some winning path $\mathcal{P}$ for Short exists in $\mathcal{G}$, then $\mathcal{P}$ does not contain v, and so $\mathcal{P}$ is also a winning path for Short in $\mathcal{G}\backslash v$. Conversely, if at the end of the game $\mathcal{G}\backslash v$ contains a winning path for Short, then so does $\mathcal{G}$, since $\mathcal{G}\backslash v \subseteq \mathcal{G}$. This means that $\mathcal{G}\backslash v$ is a Shannon game graph for Γ/F^v.

When the move $\mathfrak{m} = \text{T}^v$ is played, then new edges can be added in $\mathcal{G}$ between all pairs of neighbours of v. This creates no new winning paths for Short, since if in some final colouring a winning path contains one of the new edges, then v can be inserted into the path, removing the new edges but preserving the win for Short. Conversely, adding edges of course does not destroy any winning paths for Short either. After the neighbourhood of v has thus been turned into a clique, v itself can be removed since it is simplicial and therefore dead.[3] The result is that $\mathcal{G}/v$ is a Shannon game graph of Γ/T^v.

From the observations in Definition 2.5.1, the order in which vertices are contracted and deleted does not matter. Therefore, given any colouring $\psi \in X^{\mathcal{S}}$, all vertices in $\psi^{-1}(\text{T})$ can be contracted and all vertices in $\psi^{-1}(\text{F})$ can be deleted. The graph created by this procedure is the **reduced graph** of the colouring ψ. The reduced graph represents Γ/ψ and contains no more coloured vertices. Figure 10.4 shows an example of a Hex position and the two reduced graphs that represent it.

[3]See Section 13.2.

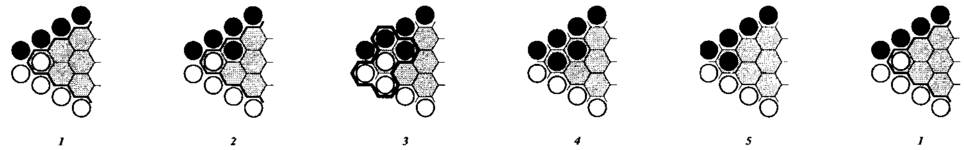


Figure 10.5: Proof of first Beck theorem.

10.5 Strategy Theorems

All previously known general theorems about Hex moves, as mentioned in Section 1.3, can be proved directly using the results from this thesis. All proofs use the fact that Hex is isotone, since the Hex outcome function is a coalition function, and some proofs use the fact that the transformation $h_n : \mathbb{X}_n \to \mathbb{X}_n$ given by

$$h_n(\xi)_{x,y} = -\xi_{y,x}$$

is an anti-isomorphism. This can be seen by noting that h_n maps directly from the black Shannon graph to the white Shannon graph and vice versa. For the remainder of this section, let $\mathbb{X}_n$ be the $n \times n$ Hex game and let MAX be White. Whenever dead cells or captured sets are used, the relevant regions are outlined and corresponding patterns can be found in Figure 13.4.

Piet Hein, 1942; John Nash, 1947: On any board size there exists a winning opening move.

This is a direct application of Corollary 5.6.2:i.

Piet Hein, 1942; John Nash, 1947: Adding a friendly piece or removing an enemy piece is never disadvantageous.

This is Theorem 5.5.3.

Anatole Beck, 1969: On any board size there exists a losing opening move [10]. The opening moves in Figure 10.5-1 and Figure 10.7-1 as well as the response in Figure 10.6-1 are losing moves.

These proofs are of the type $\text{MNX}(\mathbb{X}_n; \mathfrak{p}) \geqslant -\text{MNX}(\mathbb{X}_n; \mathfrak{p})$, which implies $\text{MNX}(\mathbb{X}_n; \mathfrak{p}) = +1$, or $\text{MNX}(\mathbb{X}_n; \mathfrak{p}) \leqslant -\text{MNX}(\mathbb{X}_n; \mathfrak{p})$, which implies $\text{MNX}(\mathbb{X}_n; \mathfrak{p}) = -1$.

Let $\psi^{(i)}$ be the colouring in Figure 10.5-i, then:

$$\begin{aligned}
\text{MNX}(\mathbb{X}_n; \psi^{(1)}, \text{MIN}) &\leqslant \text{MNX}(\mathbb{X}_n; \psi^{(2)}, \text{MAX}) && \text{(Definition 5.1.1)} \\
&= \text{MNX}(\mathbb{X}_n; \psi^{(4)}, \text{MAX}) && \text{(Corollary 5.2.2:ii, dead cell in } \psi^{(3)}\text{)} \\
&\leqslant \text{MNX}(\mathbb{X}_n; \psi^{(5)}, \text{MAX}) && \text{(Theorem 5.5.3)} \\
&= -\text{MNX}(\mathbb{X}_n; \psi^{(1)}, \text{MIN}). && \text{(Theorem 5.6.1 with isomorphism } h_n\text{)}
\end{aligned}$$

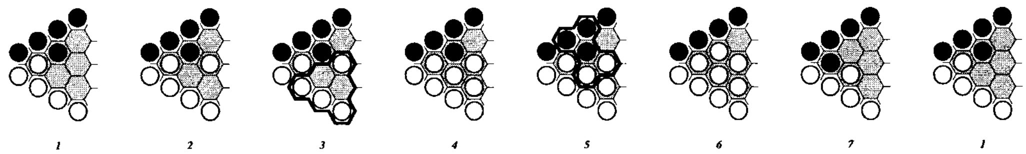


Figure 10.6: Proof of second Beck theorem.

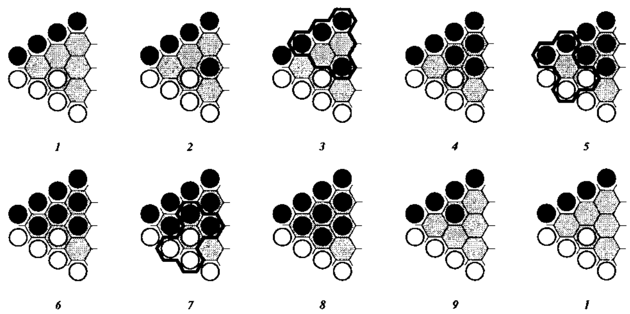


Figure 10.7: Proof of third Beck theorem.

Let $\psi^{(i)}$ be the colouring in Figure 10.6-i, then:

$$\begin{aligned}
\text{MNX}(\mathbb{X}_n;\psi^{(1)},\text{MAX}) &\geqslant \text{MNX}(\mathbb{X}_n;\psi^{(2)},\text{MIN}) && \text{(Definition 5.1.1)}\\
&= \text{MNX}(\mathbb{X}_n;\psi^{(4)},\text{MIN}) && \text{(Theorem 8.4.2, captured set in } \psi^{(3)})\\
&= \text{MNX}(\mathbb{X}_n;\psi^{(6)},\text{MIN}) && \text{(Corollary 5.2.2:ii, dead cell in } \psi^{(5)})\\
&\geqslant \text{MNX}(\mathbb{X}_n;\psi^{(7)},\text{MIN}) && \text{(Theorem 5.5.3)}\\
&= -\text{MNX}(\mathbb{X}_n;\psi^{(1)},\text{MIN}). && \text{(Theorem 5.6.1 with isomorphism } h_n)
\end{aligned}$$

Let $\psi^{(i)}$ be the colouring in Figure 10.7-i, then:

$$\begin{aligned}
\text{MNX}(\mathbb{X}_n;\psi^{(1)},\text{MIN}) &\leqslant \text{MNX}(\mathbb{X}_n;\psi^{(2)},\text{MAX}) && \text{(Definition 5.1.1)}\\
&= \text{MNX}(\mathbb{X}_n;\psi^{(4)},\text{MAX}) && \text{(Theorem 8.4.2, captured set in } \psi^{(3)})\\
&= \text{MNX}(\mathbb{X}_n;\psi^{(6)},\text{MAX}) && \text{(Theorem 5.5.2, dead cell in } \psi^{(5)})\\
&= \text{MNX}(\mathbb{X}_n;\psi^{(8)},\text{MAX}) && \text{(Corollary 5.2.2:ii, dead cell in } \psi^{(7)})\\
&\leqslant \text{MNX}(\mathbb{X}_n;\psi^{(9)},\text{MAX}) && \text{(Theorem 5.5.3)}\\
&= -\text{MNX}(\mathbb{X}_n;\psi^{(1)},\text{MIN}). && \text{(Theorem 5.6.1 with isomorphism } h_n)
\end{aligned}$$

Craige Schensted and Charles Titus, 1975: Any move that is surrounded by only three regions, and many moves that are surrounded by four or five regions, should be avoided [91]. See Figure 10.8.

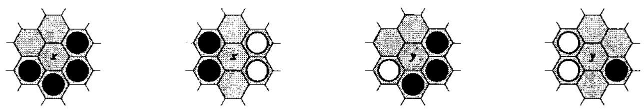


Figure 10.8: Schensted's theorems: moves marked 'x' should be avoided by both players, moves marked 'y' should be avoided by White.

Figure 10.9: Move x dominates move y for both players.

By "a region", Schensted and Titus mean either an empty cell or a Black or White string. An enumeration of the possible ways to surround a Hex cell with at most three regions shows that one of the patterns on the bottom row of Figure 13.4 must then occur.

The four-sided region in the second diagram of Figure 10.8 is one of the bottom row patterns of Figure 13.4. The third diagram, also a four-sided region, contains the one of the dominated patterns of Figure 13.4 with reversed colours. Schensted and Titus point out the dominating move that kills the center move. Both patterns are part of their "beware the square" rule.

Finally, the rightmost diagram in Figure 10.8 shows a five-sided region with a move that White should avoid. This pattern is the fifth dominated move pattern of Figure 13.4 with reversed colours.

Ryan Hayward, 2003: Any move on the second row dominates the two underlying moves on the first row. Stronger still: after a move on the second row, the two underlying cells on the first row can be "filled in" [44].

The fill-in property follows from Theorem 8.4.2 with the captured pattern second from the top in Figure 13.4. The domination property then follows from Theorem 8.5.2.

In Figure 10.9, both players should avoid move y.

This theorem is due to the author in 2003. The patterns occur in the second row from the bottom in Figure 13.4, and therefore move x dominates move y for Black and Black should avoid y. As explained in Section 13.3, White should also avoid y because it is reversible by a Black move in x.

10.6 Induced Paths

Any Hex playing computer program should be able to detect the winning condition, that being the existence of a monochrome path connecting two opposite sides of the board. One approach is to precompute all such paths.

In graph theory, an **induced path** is a set of vertices whose induced subgraph is a simple path; in other words, it is an ordered list of vertices where two vertices are adjacent in the graph if and only if they are adjacent in the list. On a Hex graph only the induced paths need to be computed, as any path contains an induced path. A winning induced path for MAX precisely corresponds to a minimal clause in the DNF formulation of the game, and a winning induced path for MIN is a minimal CNF clause. Thus the number of induced paths is the same as the number of minimal clauses.

Table 10.6.1 contains the number of distinct induced paths for MAX on rectangular Hex graphs. The "length" of the graph, namely the distance between the two borders to be connected by MAX, is listed along the vertical axis. Empirically, the number of induced pats appears to be roughly equal to $2^{cm(n-2)}$, where m is the "width" of the graph, n is the length, and c is some number usually between $\frac{1}{3}$ and $\frac{2}{3}$. Particular values for c are listed in Table 10.6.2.

Also of interest are the lengths of the induced paths. The shortest possible path on an $m \times n$ board is n; when $n \leqslant m+1$ then there are $(2m-n+1)2^{n-2}$ such paths, and fewer otherwise. Tables 10.6.3 and 10.6.4 contain information about the average path length and the maximum path length. On square graphs, where $n = m$, the average path fills about $\frac{1}{3}$ of the board.

For the 7×7 Hex graph, the total number of induced paths for one player is 68,914 with an average length of 15.63. To keep all these in memory requires storing exactly 1,077,034 cells. On the 8×8 graph the numbers are $2,195,830 \times 20.83 = 45,747,258$.

During the course of a game the number of induced paths decreases. When keeping track of the number of induced paths for MAX, any move by MIN removes all paths through the node in question. A move by MAX also removes paths, as it can generate a "short cut" in a path that is consequently no longer chordless. The game ends precisely when one of the players has no induced paths left.

Figure 10.10 shows the number of induced paths for each player during a Hex game, on a logarithmic scale. The gradual lines that start at 2 million refer to random Hex games, and it is clear that the number of paths decreases exponentially during the game. At odd move numbers the starting player has more paths than the second player, by virtue of having one more stone on the board, but at even move numbers the values are the same. The black and grey dots that start at 2 million indicate the number of paths during an actual game played between realistic players. This is based on only one such game, but it is there to indicate that realistic players likely do not deviate much from random players in this particular statistic, right until a few moves before the game is decided.

The lower lines in Figure 10.10 plot the *weighted* path counts, where a path of length l is weighted 2^{-l}. The same observations are evident as for the unweighted path count. As a side note it is remarked that the sample game was played with the swap rule and ended in a win for the second player, whose statistics are indicated by the grey dots. During the sample game the eventual winner

	1	2	3	4	5	6	7	8	9	10
1	1	2	3	4	5	6	7	8	9	10
2	1	3	5	7	9	11	13	15	17	19
3	1	5	11	19	29	41	55	71	89	109
4	1	9	25	54	107	202	370	666	1,187	2,103
5	1	16	56	148	365	881	2,082	4,808	10,900	24,420
6	1	28	124	399	1,225	3,848	12,097	36,964	109,393	315,948
7	1	49	273	1,054	3,948	16,097	68,914	288,385	1,160,865	4,551,217
8	1	86	601	2,786	12,622	65,826	382,718	2,195,830	11,948,849	62,717,006
9	1	151	1,325	7,401	40,880	273,407	2,164,772	17,117,801	126,004,636	880,801,486
10	1	265	2,923	19,712	133,828	1,158,787	12,539,626	137,083,594	1,368,020,170	12,755,638,497
11	1	465	6,448	52,514	439,378	4,956,166	73,314,006	1,105,069,149		
12	1	816	14,222	139,802	1,439,016	21,168,066	428,267,454			
13	1	1,432	31,367	372,008	4,699,144	89,908,637	2,490,235,058			
14	1	2,513	69,181	989,841	15,329,082	380,403,015				
15	1	4,410	152,583	2,634,032	50,026,736	1,608,517,375				
16	1	7,739	336,533	7,009,853	163,382,568					
17	1	13,581	742,248	18,655,329	533,773,260					
18	1	23,833	1,637,080	49,646,780	1,743,746,890					
19	1	41,824	3,610,693	132,121,693						
20	1	73,396	7,963,633	351,605,703						

	11	12	13	14	15	16	17	18	19	20
1	11	12	13	14	15	16	17	18	19	20
2	21	23	25	27	29	31	33	35	37	39
3	131	155	181	209	239	271	305	341	379	419
4	3,712	6,537	11,496	20,200	35,476	62,285	109,333	191,898	336,791	591,062
5	54,341	120,467	266,478	588,702	1,299,586	2,867,698	6,326,511	13,955,435	30,781,868	67,894,074
6	897,223	2,516,936	6,992,898	19,274,658	52,779,342	143,754,548	389,859,030	1,053,610,190	2,839,359,215	
7	17,584,658	67,283,448	255,181,419	959,469,520	3,578,534,738					
8	324,005,708	1,661,833,257								

Table 10.6.1: Number of induced paths on $m \times n$ Hex graphs for the player to traverse the graph in the vertical direction.

	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
3	1.161	1.153	1.062	0.972	0.893	0.826	0.769	0.720	0.677	0.639	0.606	0.577	0.551	0.527	0.505	0.485	0.467	0.451	0.436
4	0.792	0.774	0.719	0.674	0.638	0.609	0.586	0.567	0.552	0.539	0.528	0.519	0.511	0.504	0.498	0.492	0.487	0.483	0.479
5	0.667	0.645	0.601	0.567	0.543	0.525	0.510	0.497	0.486	0.477	0.469	0.462	0.456	0.451	0.447	0.443	0.440	0.436	0.434
6	0.601	0.580	0.540	0.513	0.496	0.484	0.474	0.465	0.457	0.449	0.443	0.437	0.432	0.428	0.423	0.420	0.416	0.413	
7	0.561	0.540	0.502	0.478	0.466	0.459	0.453	0.448	0.442	0.438	0.433	0.430	0.426	0.423					
8	0.536	0.513	0.477	0.454	0.445	0.442	0.439	0.435	0.432	0.428	0.425								
9	0.517	0.494	0.459	0.438	0.430	0.430	0.429	0.427	0.424										
10	0.503	0.480	0.446	0.426	0.420	0.421	0.422	0.422	0.420										
11	0.492	0.469	0.436	0.417	0.412	0.415	0.417												
12	0.484	0.460	0.427	0.409	0.406	0.410													
13	0.477	0.453	0.421	0.403	0.400	0.405													
14	0.471	0.447	0.415	0.398	0.396														
15	0.466	0.442	0.410	0.393	0.392														
16	0.461	0.437	0.406	0.390															
17	0.458	0.433	0.403	0.387															
18	0.454	0.430	0.399	0.384															
19	0.452	0.427	0.397																
20	0.449	0.425	0.394																

Table 10.6.2: $\frac{2 \log k}{m(n-2)}$ where k is the number of induced paths on $m \times n$ Hex graphs for the player to traverse the graph in the vertical ("n") direction.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
2	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0
3	3.0	3.2	3.4	3.6	3.8	4.1	4.4	4.7	5.0	5.3	5.6	5.9	6.2	6.5	6.9	7.2	7.5	7.8	8.2	8.5
4	4.0	4.4	4.7	5.3	6.0	6.9	7.8	8.8	9.9	11.0	12.1	13.2	14.4	15.5	16.7	17.9	19.1	20.2	21.4	22.6
5	5.0	5.6	6.0	6.7	7.8	9.1	10.4	11.6	12.9	14.1	15.3	16.6	17.8	19.1	20.3	21.6	22.8	24.1	25.3	26.6
6	6.0	6.8	7.2	8.1	9.5	11.3	13.0	14.7	16.2	17.8	19.4	21.0	22.6	24.1	25.7	27.2	28.8	30.3	31.8	
7	7.0	8.0	8.5	9.5	11.1	13.3	15.6	17.7	19.7	21.8	23.9	26.0	28.1	30.2	32.4					
8	8.0	9.1	9.7	10.9	12.8	15.5	18.3	20.8	23.2	25.5	27.9	30.4								
9	9.0	10.3	11.0	12.3	14.5	17.7	21.1	24.1	26.8	29.5										
10	10.0	11.5	12.2	13.7	16.3	20.0	24.0	27.5	30.5	33.4										
11	11.0	12.7	13.5	15.1	18.1	22.3	26.9	30.7												
12	12.0	13.8	14.7	16.6	19.8	24.6	29.7													
13	13.0	15.0	16.0	18.0	21.5	26.8	32.5													
14	14.0	16.2	17.2	19.4	23.2	29.1														
15	15.0	17.4	18.5	20.8	25.0	31.3														
16	16.0	18.6	19.7	22.2	26.7															
17	17.0	19.7	21.0	23.6	28.4															
18	18.0	20.9	22.2	25.0	30.2															
19	19.0	22.1	23.5	26.5																
20	20.0	23.3	24.8	27.9																

Table 10.6.3: Average length of induced paths on $m \times n$ Hex graphs for the player to traverse the graph in the vertical direction.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
3	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
4	4	5	6	8	9	10	12	13	14	16	17	18	20	21	22	24	25	26	28	29
5	5	6	7	9	11	13	15	17	19	21	23	25	27	29	31	33	35	37	39	41
6	6	8	9	11	14	16	18	21	23	26	28	30	33	35	38	40	42	45	47	
7	7	9	11	14	17	20	23	26	29	32	35	38	41	44	47					
8	8	10	12	16	20	23	26	30	33	37	40	44								
9	9	12	13	18	23	26	29	34	37	42										
10	10	13	15	19	26	30	34	38	42	47										
11	11	14	17	22	29	33	37	42												
12	12	16	18	24	32	37	42													
13	13	17	19	26	35	40	45													
14	14	18	21	28	38	44														
15	15	20	23	30	41	47														
16	16	21	24	32	44															
17	17	22	25	34	47															
18	18	24	27	36	50															
19	19	25	29	38																
20	20	26	30	40																

Table 10.6.4: Longest induced paths on $m \times n$ Hex graphs for the player to traverse the graph in the vertical direction.

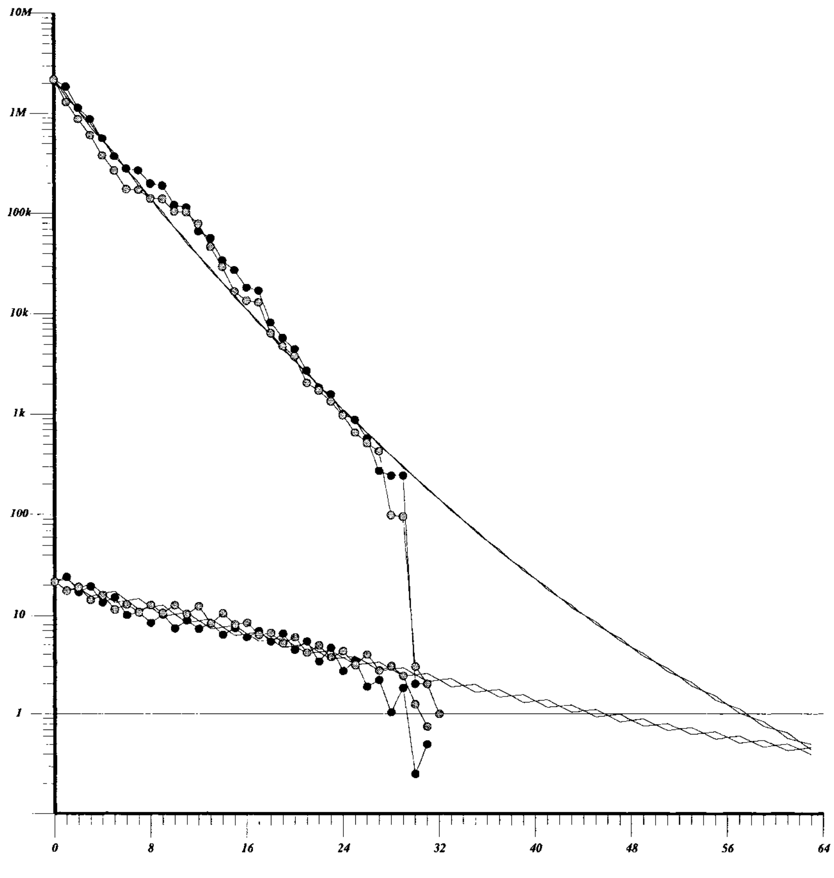


Figure 10.10: Number of paths, unweighted (high lines) and weighted (low lines), during an 8×8 Hex game. Move number is listed along the horizontal axis. Gradual lines indicate values for random players, dots indicate values for a realistic sample game, with grey dots for the eventual winner and black dots for the eventual loser.

never really appears to have a clear advantage in unweighted path count, though there are phases between moves 8 and 16 and after move 24 where grey has an advantage in weighted path count. However, this is anecdotal, as it is unknown if and when the players made any mistakes during the game.

Based on the same sample game, statistics for the average induced path length are plotted in Figure 10.11. The path lengths correspond to the size of the clauses in a CNF or DNF description of the game. All the observations about path count appear to apply to path length as well, though the winner of the realistic game appears to have had more of an advantage in path length than in path count during the game.

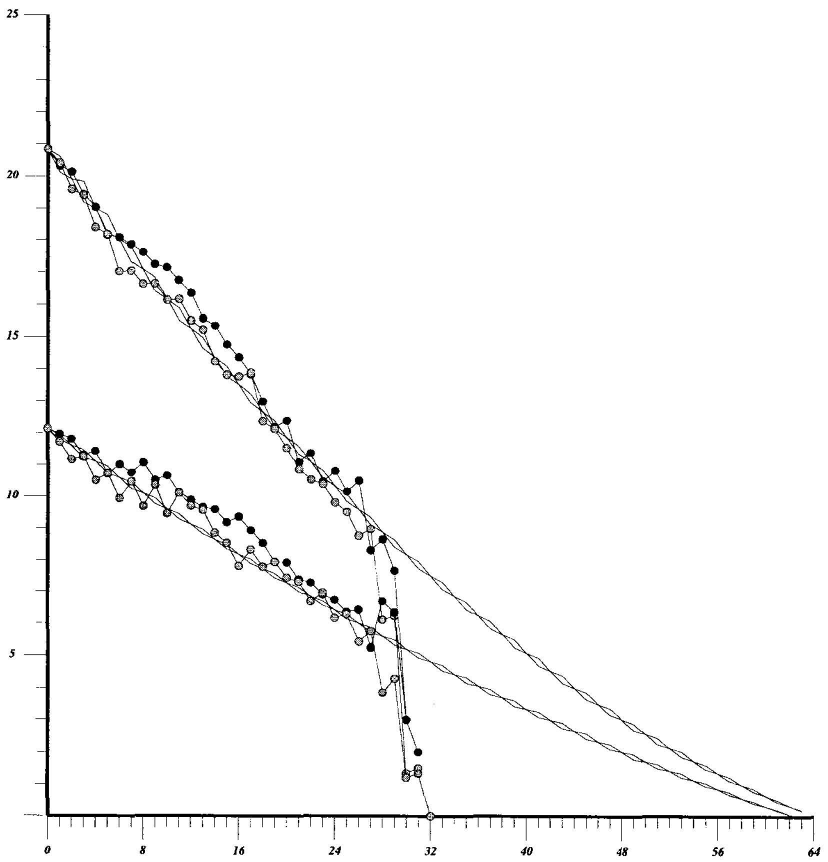


Figure 10.11: Average path length, unweighted (high lines) and weighted (low lines), during an 8×8 Hex game.

Chapter 11

Artificial Intelligence Approaches

No significant research effort or competitive results exist for GAME-SAT or set colouring games with the exception of Hex itself, but many of the techniques used in other abstract games may be applied. This chapter provides an overview of some important AI techniques known from literature in general abstract game playing, as well as Hex and QBF.

11.1 Search

The standard game tree search algorithm is known as **alpha-beta search**, given in pseudo-code in Algorithm 1. The algorithm is given a position $\mathfrak{p}$ and a **search window** $[\alpha, \beta]$. If the negamax value of $\mathfrak{p}$ falls within the $[\alpha, \beta]$ interval then the algorithm will return the correct value. If the negamax value is greater than β then a lower bound is returned, and if it is less than α then an upper bound is returned. At the root of the search tree, the algorithm is initiated with the search window $[-\infty, \infty]$.

The algorithm searches to a given maximum depth in the game tree; this depth is called the **search horizon**. If the horizon is reached and the game is not yet over, a heuristic value for the position is returned. If no moves are available in the given position, then in the case of set colouring games this means that the game is over and the outcome value can be returned.

If neither condition is met, then the algorithm recursively determines the negamax value by examining all available moves. Once the value is known to be at least equal to β, the lower bound can be returned without examining the remaining moves. This is an $\alpha - \beta$ **cutoff**, and it is these cutoffs that make the alpha-beta algorithm fundamentally more efficient than a naive recursive negamax implementation.

Consider a search tree of depth d in which each node has b children. The value b is known as the

```
ALPHABETA(p, α, β, depth)
input  : A position p with bounds α, β and a search depth.
output: The negamax value of p if in the [ α, β ] interval, a lower bound if the value is
        >β, an upper bound if the value is <α.
if (𝔐(p) = ∅) then
    /* game is over */
    return OUTCOME (p);
if (depth = 0) then
    /* search horizon reached */
    return HEURISTICEVAL (p);
result ← −∞;
foreach (m ∈ 𝔐(p)) do
    currRes ← −ALPHABETA (p ⊕ m, −β, −α, depth − 1);
    result = MAX (result, currRes);
    α = MAX (α, currRes);
    if (result ⩾ β) then
        /* α-β cutoff */
        return result;
return result;
```

Algorithm 1: The standard $\alpha - \beta$ algorithm to calculate the negamax value of a position.

branching factor. The tree contains $\mathcal{O}(b^d)$ nodes. A minimal subtree that is sufficient to prove the minimax value of the root is called a **proof tree**. It contains $b^{\lfloor d/2 \rfloor} + b^{\lceil d/2 \rceil} - 1$ as analyzed by Knuth and Moore [57]. The reason that this tree is smaller than the game tree itself is related to the fact that ideally only one child position needs to be examined in order to prove a win.

The worst case behaviour of the alpha-beta algorithm is $\mathcal{O}(b^d)$. The best-case behaviour is $\mathcal{O}(b^{d/2})$, with the proof tree limit actually achieved if the algorithm always chooses an optimal move as the first to examine. Thus the heuristics used for choosing the move ordering are of crucial importance. Given reasonable move ordering heuristics, typical performance is close to $\mathcal{O}(b^{d/2})$; in practice this means that given equal resources the alpha-beta algorithm can search a tree twice as deep as the naive negamax algorithm.

Commonly used techniques to enhance alpha-beta search include:

Transposition Tables: A hash table is used to store values, bounds, and best-move information for positions [42, 96]. This information is useful when **transpositions** occur, where the same position is reached repeatedly via different move sequences.

Iterative Deepening: The search program starts with a 1 ply horizon, then iteratively repeats the search with increasing horizons [96]. This has two advantages over a fixed depth search. First, the program becomes an "any-time" algorithm, that can be terminated at any desired moment and still return a value. Second, transposition table best-move results from shallower searches can be re-used.

Aspiration Search: The algorithm is not initiated with the $[-\infty, \infty]$ interval but a smaller window. If the returned value is in the aspiration interval then the guess was right and the value is correct. If not, then the search must be re-started with a different aspiration interval. The benefit is that searches with smaller $\alpha - \beta$ windows expand fewer nodes.

Principal Variation / Minimal Window Search: A refined and recursive variant of aspiration search. After the first move has been examined, the remaining moves are examined with a minimal-size window [65]. This leads to a very efficient refutation of all subsequent inferior moves. Whenever a move is not found to be inferior to the first move, it must be re-searched with a larger window.

Each of these techniques is intended to boost the $\alpha - \beta$ algorithm in approaching the proof tree barrier of $b^{\lfloor d/2 \rfloor} + b^{\lceil d/2 \rceil} - 1$ nodes. In theory it is possible to break this barrier because game trees can contain transpositions, enabling re-use of stored information.[1] The practical performance of game tree search programs tends to stay above the barrier due to inherent imprecisions in the move ordering heuristics.

These techniques are only described cursorily here, since they are well known in the game tree search literature, and since this thesis presents techniques specific to set colouring games that can break the proof tree barrier even when transpositions are not taken into account.

[1]This does not increase the relative efficiency of alpha-beta as compared to negamax, as the negamax algorithm profits from transpositions just as well.

Irregular Depth and Irregular Branching Factor

Game trees for many commonly played games do not have a uniform branching factor. Some existing game tree search algorithms, such as Conspiracy Number Search [66] and Proof Number Search [4], exploit this fact by essentially guiding the search towards "narrower" subtrees first, to obtain useful alpha-beta bounds quickly. These methods are not naturally suited for set colouring games, as a game tree for set colouring games always has exactly the same branching factor for each subtree, unless traces can be used.

Many game playing programs use methods for search extensions and search reductions. Search reductions, or progressive pruning, occurs when the search algorithm decides to return a heuristic value even though the horizon has not yet been reached. The reason for making such a decision might be that it is considered unlikely that the current branch will influence the values higher up in the tree. Another reason could be that the algorithm only considers the top moves as chosen by some heuristic that rates the moves; such methods are called **selective search**.

A search extension is the opposite: the search is continued even though the horizon has been reached, because the value is considered important and unsettled. Search extensions and search reductions build search trees of irregular depth. For this reason, the tree contains comparisons between heuristic values originating from different search depths. It is therefore important that such comparisons be meaningful. Some heuristics may not be suitable for these methods, such as connectivity-based heuristics in the Shannon game[2] [83]. This danger is especially present in games whose positions contain a built-in "measure of time", as do set colouring games, whereby positions at different depths in the tree are fundamentally distinguished, as opposed to games in which two move sequences of different lengths can lead to the same position.

Null Move Search

Null move search [26] is a powerful but dangerous extension of the alpha-beta algorithm. Before examining the legal moves, the null move algorithm first examines what would happen if the player to move played a **null move**, which means skipping a move. If this leads to a value at least equal to β, then β is returned immediately without examining any legal moves. The reasoning is that there must surely be moves that are better than doing nothing, so the search would generate a cutoff anyway.

Null move searches generate cutoffs in positions following a "blunder", which is a move that actually deteriorated the position for the player who made the move. In other words, a blunder is a move that is worse than a null move. Such moves are common in chess; for instance, putting a piece *en prise*. The cutoffs are generated quickly because in most implementations the null move is followed by a search depth reduction, so the overhead compared to examining the legal moves is negligible.

The danger lies in the assumption that there will be moves that are better than a null move. This is not always the case. A position in which a skip move would actually be the best possible move is

[2] See Section 12.2.

known as a zugzwang position. These positions do occur in chess, for instance. They also occur in set colouring games, but not if the game is isotone. Null moves are safe in isotone games because there are always moves that are better than a null move. However, in isotone games *all* moves are in fact better than a null move, which means that null move searches do not generate any cutoffs at all [83].

Mustplay[3] are reminiscent of this technique, as they assert that the opponent wins if the player to move does not move, that is if the opponent "plays a null move", in some specified region. However, the definition of mustplays solves the zugzwang issue essentially by allowing the opponent a star, which could be used to neutralize the null move if desired. Mustplay too, become more powerful when the *worst* move is examined first, as that tends to lead to the smallest mustplay pattern.

Threat-Based Search

Threat Space Search, later renamed db-search, was used by Allis to solve the games of Qubic and Go-Moku [3]. It identifies game-specific **threats**, where the threatening player, named the "attacker", can achieve some specified goal in one move. The search algorithm attempts to verify that the attacker can reach the goal by using threat moves only. The search is narrowly focused since the "defender" only needs to consider the moves that counter the threat.

Lambda Search, by Thomsen [100], and Abstract Proof Search, by Cazenave [22], extend this principle to the consideration of "meta-threats". A higher-order threat of level-k is a guarantee that the attacker can achieve some well-defined goal if given the opportunity to play k moves in a row, while the defender passes each time.

Thomsen's method is a game-independent way to build a tree to determine whether the attacker has a level-k win, which is a win using only level-k threats. Any tree node where the attacker to move has a lower-level win returns a win value. Any node where the attacker to move has no level-k threat moves returns a loss value. If a level-k win is found, then the attacker can achieve the goal. If the level-k search returns a loss, then it is not yet known whether the attacker can achieve the goal, and a higher-order search is started at the root.

Cazenave's Abstract Proof Search is a very similar approach, in which low order threats are identified by game-specific knowledge. In the case of set colouring games a level-k threat can be identified easily if the outcome function is given in CNF or DNF form. If it is in CNF then MIN has a level-k threat if there exists a clause with at most k variables. For a DNF formula the same holds for MAX.

Lambda search and Abstract Proof Search have similarities with null move search, in that a higher-order threat is defined in terms of null moves played by the defender. But the attacker does not use null moves, and search depth is not reduced following a null move. As with null move searching, lambda searching can have problems with zugzwang.

Threat-based methods do well when the game allows for long but "narrow" meta-threat sequences,

[3] See Section 9.2.

focusing the search by examining only moves that are relevant to the threat. They build trees of highly irregular depth and rely on well-defined goals; they do not combine well with heuristic values. Dynamic trace search, to be described in Section 11.2, similarly considers only the moves that are relevant to the threat.

Monte Carlo Search

A somewhat counterintuitive selective search method is Monte Carlo search, where the move ranking heuristic involves a degree of randomness. Stochastic search methods are more established in games featuring a stochastic element in the rules, such as Backgammon [99], or featuring imperfect information, such as Scrabble [93, 95] and card games [16], where exploration of a representative sample of the search space is enforced by the probabilistic nature inherent to the game.

However, stochastic methods have also been used in deterministic games of perfect information. Grigoriev tried a simulated annealing approach for the games of Go-Moku and Renju [43], where moves were evaluated according to game-specific knowledge and then pruned stochastically, where lower rated moves had a higher probability of being pruned. The process was controlled by an annealing temperature; as the temperature was lowered gradually, lower ranked moves were more and more likely to be pruned.

Brügmann incorporated similar ideas into the game of Go [19]. Feeling that Go search trees would be too large even to consider just two continuations at each search node, Brügmann instead applied annealing to the game as a whole. All the moves on the board were ranked statically, and then a game was simulated where the players at each move chose a random move, with higher ranked moves more likely to be chosen. At the end of the game, move rank was increased for *all* moves played by the winner, and decreased for all moves played by the loser. The whole process was then repeated numerous times, with gradually decreasing temperature.

This method benefits from the observation that, in the game of Go, a good move tends to be good whenever it is played. This is closely related to the fact that the same moves played in a different order will often lead to the same position. In Go this is not always true due to captures, but it *is* perfectly true for all set colouring games. Note that even with perfect move order independence the static quality of a given move still changes during the course of a game, as the context of the board position changes.

The move order independence is also exploited in more recent work by Bouzy and Helmstetter [17], also for the game of Go. All initial moves are scored by playing out the rest of the game randomly, using minimal Go knowledge. All moves played by the winner receive credit for the win, referred to by Bouzy and Helmstetter as "all moves as first". Another method used in their experiments is "progressive pruning", where after a number of simulations are performed the lower scoring moves are no longer simulated.

When the rest of the game is played out randomly the generated search "tree" has a branching factor of 1, which would be more accurately described as a heuristic position evaluation rather than a search. Stochastic evaluation will be discussed in Section 12.5. Stochastic search methods are used when the full game tree is too large to be searched, and the static position evaluation is more unreliable than a Monte Carlo playout of the rest of the game. Even though random moves are unlikely to be good moves, the playout still gives useful information since *both* simulated players are playing equally weakly. Recent progress by Coulom in the game of Go has involved Monte Carlo search in which move choice are biased towards moves that have accumulated more profitable statistics [24]. In this approach the search will converge to the correct minimax result for the part of the search tree that can be kept in memory.

11.2 Dynamic Trace Search

The dynamic trace search method is a generalization of a search enhancement for Hex first presented by the author at the Computers and Games workshop of the 2000 Computer Olympiad, and later specified in a tech report [84]. Pseudocode is listed in Algorithm 2. In isotone set colouring games a dynamic trace can be seen as a guarantee that the attacker wins even if the defender plays *all* rational moves outside of the dynamic trace at once, ensuring that the attacker certainly wins if the defender plays only one such move. For non-isotone games the assertion would be somewhat weaker: The attacker wins even if the defender plays any *even* number of moves outside the dynamic trace at once.

Dynamic traces in Hex have been crucial in solving all 7×7 openings [46], where they were used to identify mustplay regions. As mentioned in Section 11.1, null moves do not generate any α-β cutoffs in Hex. Yet they are very effective at discovering mustplay regions. The reason is that mustplay regions tend to be smaller after weaker moves by the opponent, as weaker moves allow for "easier" wins, so the smallest mustplay region occurs after the move that is guaranteed to be the weakest of all in Hex, namely a null move. This reasoning applies only to isotone games, as a null move in a game Γ is really a move in the star of $\Gamma*$, which may have a different outcome is the game is non-isotone.

Dynamic trace search in Hex is closely related to and based on a proof method used by Yang,[4] capable of automatically discovering dynamic traces. Yang's patterns were all devised by hand, and no efficient method of discovering or using those patterns algorithmically has been proposed yet. However, Yang's patterns are far more powerful and economical, since they establish *local* connections that can be transposed and re-used in other areas of the board, and since they are capable of decomposing a connection into sub-connections that can be played independently. It is the decomposition that plays a powerful role in Yang's patterns and in Anshelevich's virtual connections.[5]

The dynamic traces discovered by Algorithm 2 are global and do not decompose. In [84] a method was proposed to decompose such patterns algorithmically during a search. This method is Hex-specific, or more generally specific to the Shannon game, as it takes advantage of the graph structure of the game board. Section 14.3 contains a description of the proposed algorithm.

[4]See Section 11.5.
[5]See Section 11.5.

```
PatternAlphaBeta(p, α, β, depth)
input  : A position p with bounds α, β and a search depth.
output: A variable of the type (value, pattern), with −(value, pattern) =def (−value,
        pattern). The component value is the negamax value of p if in the [ α, β ]
        interval, a lower bound if the value is >β, an upper bound if the value is <α.
        The component pattern is a dynamic trace pattern for p.

if (𝔐(p) = ∅) then
 └ return (Outcome (p), ∅);

if (depth = 0) then
 | /* heuristic value must be in interval (−1,+1) exclusive */
 | /* dynamic trace pattern is irrelevant */
 └ return (HeuristicEval (p), ∅);

result ← (−1, ∅);
mustplay ← 𝒰(p);

foreach (v ∈ mustplay) do
 | m ← a rational move χ^v;
 | currRes ← −PatternAlphaBeta (p ⊕ m, −β, −α, depth − 1);
 | if (currRes.value = +1) then
 |  └ return (+1, v ∪ currRes.pattern);
 | else if (currRes.value = −1) then
 |  | result.pattern ← result.pattern ∪ currRes.pattern;
 |  └ mustplay ← mustplay ∩ currRes.pattern;
 | else
 |  | /* heuristic value obtained, pattern is irrelevant */
 |  | result.value = max (result.value, currRes.value);
 |  | α = max (α, currRes.value);
 |  | if (result ⩾ β) then
 |  |  | /* α-β cutoff */
 |  |  └ return result;
 └  └

return result;
```

Algorithm 2: Dynamic Trace search algorithm to calculate the negamax value and a dynamic trace pattern of a position in an isotone game.

11.3 Heuristics

Heuristics generally come into play in two different ways in game playing programs: leaf node evaluation, and move ordering. **Move ordering** involves selecting which move to try first; a process that is of crucial importance even when solving a position is feasible, since it determines the size of the generated tree. Many move ordering methods are game-specific. Well studied game-independent move ordering methods include:

Killer Moves: If a move generates an $\alpha - \beta$ cutoff, it is tried first in sibling nodes as cutoffs in siblings tend to be caused by the same strategic reasons.

History Heuristic: A global history table keeps track of the number of cutoffs generated by each possible move. Moves with a high history value get higher priority.

Refutation Table: A separate history table is kept following each possible move. This enables the detection of refutations that are specific to the opponent's mistake.

See [89, 90] for more information on these heuristics.

Leaf node evaluation refers to the estimate of the game-theoretical value of a position, which is often taken as an estimate of the "chances" of winning the game starting from the position in question. This becomes relevant when it is not possible or feasible to solve a position perfectly. Leaf node evaluation tends to be more game-specific than move ordering. Yet there are some general methods that are applicable to wide classes of games:

Material count: The heuristic value of a position is some linear function of the pieces left on the board. The weights of the linear function can be subject to machine learning. Overwhelmingly the most important ingredient in chess evaluation, material count applies only to games that involve the capture of pieces and is entirely irrelevant in set colouring games.

Mobility: The number of legal moves available to a player has proven to be correlated to winning chances in games such as chess.[6] This too is not relevant in set colouring games, where the mobility at each stage of the game is fixed ahead of time, entirely independent of what happened previously.

Monte Carlo evaluation: Apparently applicable to any game whatsoever, Monte Carlo evaluation involves playing out the remainder of the game using entirely random moves for both players.

Monte Carlo evaluation gained prominence in games involving a stochastic element, such as dice rolls, where **rollout analysis** can be used to evaluate a board position. This method plays out the remainder of the game a given number of times, and averages the outcome. This would yield the mathematically exact evaluation of a position if the choice of moves within each simulation were

[6] A striking example of this was the discovery that a heuristic value that returns a *random* number nevertheless produced stronger chess as the search depth increased [9]. It was then understood that this method essentially rewards positions with high mobility.

optimal. However, in practice it turns out that even with suboptimal move choice the rollout analysis provides very accurate information, as long as the move choice is "equally bad" for both players during each simulation. Results of this kind are known for backgammon [99] and Scrabble [93, 94, 95], for instance.

For games that do not feature a stochastic element, Monte Carlo evaluation methods can still be used. One can play out the game a number of times, enforcing some degree of randomness in order to ensure variety in the simulations. Abramson introduced the **expected outcome model**, testing it in chess [1] and tic-tac-toe and Othello [2]. It is a game-independent metric where the games were played out randomly. This approximates the probability that the player to move would win the game if both players played randomly. This differs from the outcome with optimal play; however, there is a degree of correlation. Despite the random behaviour the model produces useful information, since both players play equally badly within each simulation.

In set colouring games there is a direct correspondence between Monte Carlo evaluation and a variant of mobility count, if mobility is redefined as the number of winning sets remaining for each player. Section 12.5 will continue on this topic.

11.4 State of the Art in QBF

Where Boolean Satisfiability (SAT) has been well studied for many years, QBF has only started attracting attention in recent years, with a QBF solvers competition as part of the annual SAT conference [61, 60, 70]. A QBF problem can be seen as a game where the "existential player" tries to satisfy the formula while the "universal player" attempts to falsify it. Yet QBF differs from GAME-SAT and set colouring games in that the order of assignment of variables in QBF is fixed; the only choice faced by the players at each stage is a binary one.[7]

Since QBF is PSPACE-complete there exists a reduction from any GAME-SAT or set colouring game to a QBF instance. The difference is in economy of representation. It should indeed be noted that any QBF instance can in fact be expressed as a regular SAT problem, but at the expense of a possibly exponential increase in problem description size.

Given the fixed order of assignment in QBF, most of the work done by QBF solvers involves the early detection of a satisfiable or unsatisfiable instance before all variables have been assigned:

Contradictory clause: If there is a clause containing only universal literals, or containing no literals at all, the formula is not satisfiable.

Trivial truth: If removing all universal literals yields a satisfiable formula, the entire formula is satisfiable.

[7]There can be some leeway, since two quantifiers of the same type may exchange places, and since $\exists_x[\forall_y[f(x,y)]] \implies \forall_y[\exists_x[f(x,y)]]$.

Trivial truth is a special case of an optimal colouring obtained from a punctuated formula. Other heuristics involve the "forced choice" of a value to be assigned to a literal.

Unit literal: An existential literal occurring in a clause with no other existential literals, and all other literals in the clause to be assigned later.

Pure literal: A variable occurring only in negated form or only in unnegated form.

The unit literal and pure literal heuristics are special cases of superrational moves. A large part of the QBF solver literature focuses on data structures to keep track of pure literals, unit clauses, and void quantifiers [39], as well as checking for subsumed clauses.

Search-based programs typically use **Conflict and Solution Directed Backjumping** (CSBJ), which involves search tree pruning based on conflict sets and solution sets.

Conflict set: A set of existential variables that causes a contradictory clause.

Solution set: A set of universal variables such that all clauses not satisfied by the existential assignment are satisfied by at least one of the universal variables.

A cutoff is generated when the variable to be assigned is not contained in such a set. These concepts are closely related to optimal colourings. The detection of such sets can be subject to learning [41].

The strongest QBF solvers employ CSBJ search-based programs [40]. QBF search can be enhanced with stochastic search techniques [38]; this combination is currently considered the state of the art [70]. The size of instances that can be solved ranges from at least several hundred to as much as a million variables and clauses [61]. Though it is not clear how a game like Hex would be best encoded as a QBF instance, the number of clauses would correspond to Hex in the board size range between 5×5 and 8×8.[8]

The QBF solver competitions involve benchmark instances that were taken from practical problems, such as circuit design, as well as randomly generated problems. They do not generally involve problems taken from actual games. A first initiative towards game-specific applications was given by Zhao and Müller with GAME-SAT, where some heuristics involving move ordering were explored [107].

All work in QBF and GAME-SAT has thus far focused on *solving* instances; no attention has yet been given to heuristic methods to play well in instances whose perfect solution is infeasible. One particular problem with QBF solving competitions has been that the very nature of PSPACE-completeness prevents efficient verification of the results. The QBF benchmarks that come from underlying problems may have solutions that can be derived by other means, from insight into the structure of the problem, but for randomly generated QBF instances this is not possible. A "QBF playing competition" where programs play against each other by actually assuming the roles of the existential and universal player would circumvent this problem.

[8]See Section 10.6.

11.5 State of the Art in Hex

The state of the art in Hex can be split into two areas: Explicit winning strategies for positions that can be solved perfectly, and strong heuristic play for positions that cannot yet be solved. This dichotomy is present because the strongest known results for explicit winning strategies have been devised by hand, not discovered by computers.

The strongest known Hex playing programs are based on Anshelevich's virtual connection method [5, 6, 7]. The 2003 and 2005 Computer Olympiad tournaments have been wins for the program *Six* by Melis, with *Mongoose* by Hayward *et al* taking the silver [68, 69]. Both programs are based on virtual connections. In the estimation of the authors in question, the programs are roughly on par with the strongest human players on boards of size 9×9 and 10×10 [67]. On larger board sizes the human players have the upper hand.

On 6×6 and smaller boards the computers can play perfectly; an extensive opening library for 6×6 was published online on the Queenbee web pages [81]. Yang published the first explicit winning strategies for some of the 7×7 and 8×8 openings [104, 105, 106]. These strategies were devised by hand. Noshita provided an updated method that allows for a more economical representation of the proofs [74].

The proofs provided by Yang have been translated by Hayward *et al* into a notation system that allows the proofs to be verified by computer [45]. The notation system is based on proof trees which are reduced considerably in size and complexity. A **proof tree** is a tree where each move by the winner specifies all the opponent's replies, while each move by the loser specifies just one winning reply. Proof trees for Hex tend to contain many identical subtrees. Identical subtrees are merged into one tree, where the root node represents a choice within a certain collection of moves. The resulting trees are called **excised trees.**

Excised trees are then simplified further when the strategy partitions into independent sub-strategies, essentially partition strategies where the winner need only respond to the partition region in which the opponent just moved. Figure 11.1 shows such an **autotree,** so called because it contains only the moves for the winning player.

The recipe for using an autotree is as follows. Each labelled node represents a move to be played by the winning player. The winning player plays the move at the root of the tree. Whenever the opponent plays a move, represented by an unlabelled node, the winner selects a subtree that does not contain this move. Such a subtree is guaranteed to exist, by the autotree property that the common intersection of all subtrees of a given unlabelled node be empty.

If a labelled node contains more than one unlabelled child, then the winner must subsequently play *all* resulting subtrees simultaneously. The autotree property guarantees that any opponent's move requires a reply in at most one subtree.

When all active strategy trees have been reduced to a leaf node, the existence of a winning path is guaranteed. This comes from the second property that autotrees must satisfy, namely if one arbitrarily removes all but one child of every *unlabelled* node then the collection of remaining labelled

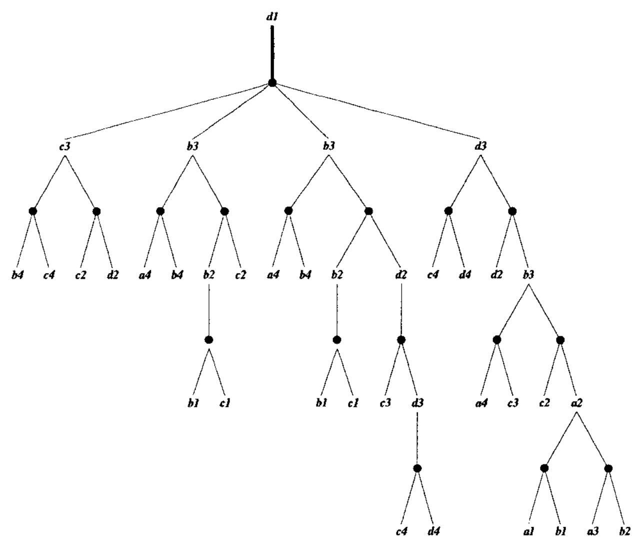


Figure 11.1: A 54-node autotree fully describing a winning strategy for 4 × 4 Hex after the opening move d1.

nodes must contain a winning path.

In the example of figure 11.1, if the opponent's reply to d1 is anything other than b4, c4, c2, or d2, then the winner responds with c3. Subsequently the winner plays both resulting subtrees, responding to b4 with c4 and vice versa, and responding to c2 with d2 and vice versa. Figure 11.1 illustrates the economy of strategy representation that this method allows, using only 54 nodes where the proof tree of the same strategy contains 7104 nodes.

Autotrees can be simplified further by encoding frequently occurring patterns into **macros** that may be re-used in translated, rotated, and mirrored form anywhere on the board. Hayward *et al* thus translated Yang's proof into autotree macros and verified the proofs by computer. The full proof required 7333 nodes. One gets an idea of the economy of representation when one considers that the full proof tree must be at least 13 ply[9] deep, and the first 13 ply alone contain some $12.7 \cdot 10^9$ nodes already.

Much progress could be made if such autotree macros could be discovered and used automatically. No methods have yet been proposed to this end. The dynamic traces described in this thesis are closely related to Yang's proofs and excised trees, in that they merge the subtrees for opponent's moves that have identical replies. Indeed the autotrees implicitly specify dynamic traces; if one marks all the cells corresponding to labelled nodes descending from a given node, the result is precisely a dynamic trace. The difference is that Yang's proof describes *local* patterns that can be turned into macros and re-used elsewhere on the board, while dynamic traces are global structures.

[9] A **ply** is one level in a game tree.

Chapter 12

Shannon Game Heuristics

In AI game playing engines two main kinds of heuristics are used: board evaluation, and move evaluation. A board evaluation estimates the winning chances given a position; a move evaluation estimates the strength of a given move in a given position. The two can be related, since a move evaluation could be delivered by calculating the board evaluation of the resulting position, and a board evaluation can be obtained by picking the most advantageous move evaluation. But there often are considerations of implementation efficiency that will lead to both types of evaluations being used at different points in the same program. Sometimes a third heuristic may be used, namely a direct move heuristic, which picks a move outright without comparing heuristic values.

Since the Shannon game is played on a graph, the game has an extra layer of structure beyond being an isotone set colouring game. In particular, the graph imposes notions of distance and locality. This chapter describes various heuristics that have been used or proposed for Hex, and that can all be generalized to the Shannon game.

12.1 Flow

The goal of connecting two terminal vertices in a graph is reminiscent of network flow models. One may think of the game graph as a network through which fluid or electrical current is to flow between the terminals. The earliest mention of this idea is by Shannon himself, who described building a physical Hex playing machine using this approach in 1953 [92]. Shannon defined a two-dimensional potential field, with black and white pieces and goal areas as opposite charges. An electrical resistance network was built that allowed locating "certain specified saddle points" [34], where the next move would be played. Later Shannon constructed a similar machine to play the game Bridg-It,[1] a game which he called "Birdcage". Graph edges were represented by resistors which were removed or short-circuited according to moves by Cut and Short. Moves were selected

[1] See Section 4.5

by picking the resistor with the highest current [36].

An algorithmic version of the flow model was built by Anshelevich for his Hex playing program *Hexy* [5, 7]. Anshelevich uses a variant of the model where the edges of the game graph contain resistors, and the heuristic evaluation of the board position is deemed to be inversely related to the electrical resistance between the terminals. This is equivalent to a flow model, where the flow capacity of an arc is the inverse of its resistance. When an enemy piece is played, the electrical wires attached to it are cut, which corresponds to reducing the flow capacity to zero. When a friendly piece is played, the resistors are removed from the wires attached to it, corresponding to increasing their capacity to infinity. Hexy calculates the energy dissipation at each node to arrive at a heuristic move evaluation.[2]

The important addition that Anshelevich developed consists of **virtual connections**. The concept of virtual connections has been recognized implicitly by Hex players from the inception of the game, and is described explicitly earlier by Berge in 1977 [11]. A virtual connection is defined as follows.

Definition 12.1.1. Let $\mathcal{G}$ be a graph and let $\mathcal{T}, \mathcal{T}' \subseteq \mathcal{V}(\mathcal{G})$ be two collections of nodes. Let $\psi \in \mathbb{T}^{\mathcal{V}(\mathcal{G})}$, and let $\mathcal{S}' \subseteq \psi^{-1}(\phi)$. If player $\mathfrak{c} \in \mathfrak{C}$ has a second-player strategy that ensures connecting $\mathcal{T}$ and $\mathcal{T}'$, using only nodes from $\mathcal{S}'$, then $\mathfrak{c}$ has **a strong virtual connection** between $\mathcal{T}$ and $\mathcal{T}'$ in ψ, with **carrier** $\mathcal{S}'$. If $\mathfrak{c}$ has a first-player strategy that ensures this, then $\mathfrak{c}$ has **a weak virtual connection** between $\mathcal{T}$ and $\mathcal{T}'$ in ψ. A virtual connection, strong or weak, is denoted as $\mathcal{T} \xleftrightarrow{\mathcal{S}'} \mathcal{T}'$. Since a strong virtual connection also meets the criteria for a weak virtual connection, the adjective *strong* may optionally be omitted.

> *Observation i:* Since the Shannon game is isotone, a strong virtual connection also meets the criteria for a weak virtual connection.

The specification of the carrier comes from the crucial observation that a virtual connection typically does not need to use all of the uncoloured nodes in the graph. The carrier is the virtual connection equivalent of a dynamic trace.

A virtual connection need not be between the two terminals of the game graph. Virtual connections can be identified between other pairs of nodes or groups of nodes as well. A virtual connection between the two terminals will be called **a global virtual connection**; any other virtual connection is **a local virtual connection**. Based upon this, Anshelevich describes two rules with which local virtual connections can be combined to form bigger virtual connections.

Definition 12.1.2. Let $\mathcal{G}$ be a graph and let $\psi \in \mathbb{T}^{\mathcal{V}(\mathcal{G})}$. Consider two virtual connections $\mathcal{T}_0 \xleftrightarrow{\mathcal{S}_0} v$ and $v \xleftrightarrow{\mathcal{S}_1} \mathcal{T}_1$ for player $\mathfrak{c}$ in ψ, where $\mathcal{S}_0 \cap \mathcal{S}_1 = \varnothing$. If $\psi(v) = \lambda(\mathfrak{c})$ then $\mathcal{T}_0 \xleftrightarrow{\mathcal{S}_0 \cup \mathcal{S}_1} \mathcal{T}_1$ is a virtual connection for $\mathfrak{c}$ in ψ. If $\psi(v) = \phi$ then $\mathcal{T}_0 \xleftrightarrow{\mathcal{S}_0 \cup \mathcal{S}_1} \mathcal{T}_1$ is a weak virtual connection for $\mathfrak{c}$ in ψ. This is called the **AND rule.**

Definition 12.1.3. Let $\mathcal{G}$ be a graph and let $\psi \in \mathbb{T}^{\mathcal{V}(\mathcal{G})}$. Consider a set of weak virtual connections $\{\mathcal{T} \xleftrightarrow{\mathcal{S}_i} \mathcal{T}'\}_{i \in \mathbb{Z}_k}$ for player $\mathfrak{c}$ in ψ. If $\bigcap_{i \in \mathbb{Z}_k} \mathcal{S}_i = \varnothing$ then $\mathcal{T} \xleftrightarrow{\cup_i \mathcal{S}_i} \mathcal{T}'$ is a strong virtual connection

[2] In electrical network theory, the energy dissipation in a resistor of resistance R at a current of I is equal to I^2R.

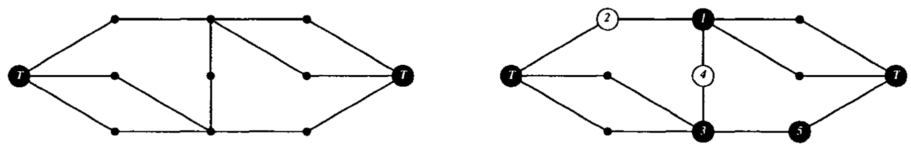


Figure 12.1: A weak virtual connection (left) that cannot be proved with the And-Or rules.

for $\mathfrak{c}$ in ψ. This is called the **OR rule.**

The intuition behind the AND rule is that if a player $\mathfrak{c}$ can connect $\mathcal{T}_0$ to v and v to $\mathcal{T}_1$, and the connections do not interfere with each other, then $\mathfrak{c}$ can connect $\mathcal{T}_0$ to $\mathcal{T}_1$. The OR says that if $\mathfrak{c}$ has several ways of connecting $\mathcal{T}$ to $\mathcal{T}'$ when going first, and the opponent cannot interfere with all of them at the same time, then $\mathfrak{c}$ has a way of connecting $\mathcal{T}$ to $\mathcal{T}'$ when going second.

These rules enable Anshelevich to build up larger virtual connections starting from the smallest "atomic" virtual connections, namely the ones with empty carriers, connecting two groups that are already connected. In any colouring it is guaranteed that either both players have a weak global virtual connection, or one player has a strong global virtual connection. Finding a global virtual connection must therefore be PSPACE-hard. Indeed Kiefer proved that it is PSPACE-complete in 2003 [54, 55].

Unfortunately this method is not guaranteed to find a global virtual connection at all, as pointed out by Anshelevich [5]. Figure 12.1 shows an example, based on a diagram given by Anshelevich, of a virtual connection that cannot be reduced to smaller virtual connections using the AND-OR rules. Black has a weak virtual connection between the two terminals. The only way to achieve this is, without loss of generality, to play at the vertex marked ❶ in the diagram on the right. The AND-OR rules would then attempt to prove the connection by finding strong virtual connections between ❶ and both of the terminals. Yet there actually is no virtual connection at all for Black between ❶ and the terminal on the left.

The deeper reason that the AND-OR rules cannot reduce this connection is that the AND rule contains the implicit assumption that the connection will use the intermediate vertex, being ❶ in the case of Figure 12.1. However, when play proceeds as indicated in the diagram on the right, where all of Black's replies are forced, Black does establish a connection between the terminals but not through ❶.

The deduction rules were extended by Rasmussen and Maire to be able to find all virtual connections [79]. Their method proves the "tricky" virtual connections by doing a local game tree search that uses the defender's mustplay region.[3] It is not yet known how effective this method is in practice.

[3]See Section 9.2.

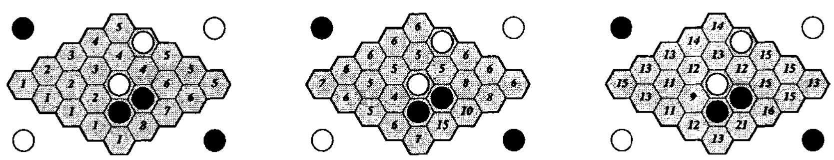


Figure 12.2: Examples of the two-distance applied to Hex: white distance to lower white edge (left), white potential (center), total potential (right).

12.2 Connectivity

Network flow models are related to the concept of graph connectivity, which refers to the number of distinct paths that connect two vertices in a graph. A high degree of connectivity is correlated with a high flow capacity, which in Hex leads to a favourable position as there are many ways to connect.

In general, a very good property for a heuristic would be to implicitly recognize when a goal has already been reached. The AND-OR rules do not have this property. An evident heuristic in the Shannon game that does accomplish this is the graph distance between the terminals, since the distance equals zero when Short connects and infinite when Cut disconnects. Yet the graph distance is more naturally suited to puzzles than to games, since it implicitly assumes that one can always choose the most advantageous route. It amounts to counting the number of "free moves" one would need to complete a connection, ignoring the opponent's thwarting endeavours.

A modification of graph distance to be used in an adversarial search environment was introduced by the author and used in the Hex playing program *Queenbee* [82, 83]. The distance of a node to the goal according to the standard graph distance is one more than the smallest distance of its neighbours to the goal. In the **two-distance** this is replaced with the *second* smallest distance of its neighbours. The motivation behind this choice is that the opponent may block the shortest path, thus it is advantageous to have a good second-shortest path available.

Some examples of the two-distance are shown in Figure 12.2. Note that the distances are calculated in the *reduced graph* of the position. The diagram on the left gives the two-distance to the lower left edge from White's point of view, that is, with white nodes contracted and black nodes removed. The middle diagram gives the sum of the White two-distances to the two white edges, indicating which empty cell is closest to being connected to both sides. These numbers are called the **white potentials** of the empty cells. The black potentials are calculated between the two black edges and from Black's point of view. The diagram on the right gives the sum of the black and white potentials; nodes with a low total potential tend to be good move choices since they are important in either establishing a friendly connection or blocking an enemy connection [82].

It can be appreciated that the two-distance is more suited than the regular distance to Hex in particular by considering that placing a stone on an empty board does not decrease the opponent's regular distance at all, regardless of where the stone was placed, whereas it does decrease the opponent's two-distance if the stone was placed near the center. More importantly, the two-distance cannot percolate through a "two-bridge", which consists of two enemy pieces with two empty mutual

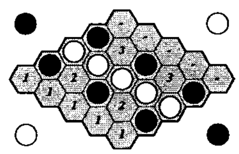


Figure 12.3: A false positive for the two-distance: the white distance is infinite, yet White wins.

neighbours. A two-bridge forms a virtual connection, and the two-distance implicitly recognizes this.

In addition to giving correct answers in decided positions, another important property of a heuristic is to return "goal reached" *only* when a goal has indeed been reached or can forcibly be reached. Unfortunately the two-distance fails in this regard. Figure 12.3 shows a position in which the white two-distance between the white borders is infinite, yet white wins. The Queenbee program uses the lowest black and white potentials in its board evaluation, which would still return a finite answer for White in Figure 12.3, but from this example larger positions can easily be constructed where all white cell potentials are infinite and yet white still wins.

The merit of the two-distance is that it rewards having two short connections higher than having just one short connection. A related idea would be to measure the normal distance, but have the heuristic incorporate the *number* of available paths of a given length. Such heuristics will be described in Section 12.4.

12.3 Y-Reduction

Schensted's Y-reduction technique, described in Section 4.6, leads to a heuristic that is entirely unique to Y. However, since Hex is a special case of Y, it can be used for Hex as well. The method does recognize reached goals and never gives false positives.

Y reduction is based on the fact that exactly three cells meet at every intersection, defining a unique colour for the corresponding node on the next smaller board. The author has proposed extending this method to apply to partially filled boards by using a probabilistic approach [84]. Each cell is assigned a "probability of ownership", with initial probabilities being 0 or 1 according to the owner of a cell, and $\frac{1}{2}$ for empty cells. When reducing a triangle of cells with probabilities p_1, p_2, and p_3, the probability q of owning the reduced cell is the probability of owning at least two of the original three cells. According to probability theory this leads to

$$q = p_1p_2 + p_1p_3 + p_2p_3 - 2p_1p_2p_3.$$

Y is not a game of chance. Moreover, the probabilities p_i are not independent, since playing a piece on the board alters a growing number of probabilities down the chain of reduced diagrams. Nevertheless this method may give a good heuristic indication.

To simplify matters, the interval $[-1, +1]$ can be used instead of $[0, 1]$. In that case, already played pieces have values -1 and $+1$ and an empty cell has value 0. The equation then becomes

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

This reduction method generates a pyramid of values, starting with the $\frac{1}{2}n(n+1)$ cells of the size-n game board and going down to the single value of the size-1 board that represents the final evaluation. The number of calculations carried out in the entire reduction chain is $\frac{1}{6}n(n+1)(n+2)$. The calculations can be done incrementally, since they are all local. This saves quite a bit of work as playing a move leaves most of the values unchanged on the bigger boards in the reduction chain.

The reduction heuristic can be used to rate the available moves. The most straightforward way to do this is to try each move and see how much the evaluation changes. This would amount to a $\mathcal{O}(n^3) \times \mathcal{O}(n^2)$ computation. A good estimate can be obtained much faster by calculating the partial derivative of the final evaluation with respect to each of the values in the reduction pyramid. These can be calculated easily; if v is the final evaluation then

$$\frac{\partial v}{\partial p_1} = \frac{\partial v}{\partial q} \cdot \frac{\partial q}{\partial p_1} = \frac{\partial v}{\partial q} \cdot \frac{1}{2}(1 - p_2 p_3).$$

This is the contribution of $\triangle(p_1 p_2 p_3)$. Since p_1 is usually part of three reduced triangles, the contributions of the other triangles need to be added. This computation builds a second pyramid of $\mathcal{O}(n^3)$ values in $\mathcal{O}(n^3)$ steps, using the values contained in the reduction pyramid. The move evaluation pyramid is built in the other direction, starting with the size-1 diagram.

Applying this Y method to the game of Hex meets an ideological obstacle from the outset, since a Hex position can be encoded as a Y position in two different ways: by extending the top white edge and the right black edge, as in Figure 4.6, or by extending the other two edges. Applying the Y reduction heuristic to these two encodings gives two different answers. Tests of this heuristic showed no promising results and were abandoned by the author. The only merit of the method is that it flags no false positives and no false negatives. A better heuristic that achieves this is based on counting paths, to be described in the next section.

12.4 Counting Paths

Due to the considerations of the previous sections, if a Shannon game heuristic is to use graph distance it must consider not just the path length but also the number of paths. There is an algorithm by Kloks and Kratsch that finds all minimal separators between two vertices in a graph $\mathcal{G}$ in $\mathcal{O}(|\mathcal{V}(\mathcal{G})|^3 n)$ time, where n is the number of minimal separators [56]. A minimal separator in their terminology is a set of vertices that disconnects two given vertices, with the property that no subset achieves the same. This corresponds to a minimal colouring in the Shannon game, and in Hex it corresponds to an induced path in the opponent's Hex graph.

When a move is played in a certain cell, all induced paths in the opponent's Hex graph through that cell are blocked. Since the goal of the game is equivalent to reducing the number of induced paths in the opponent's pass to zero, the induced path count also leads to a move heuristic by considering the number of induced paths through each cell. Note that playing a move also reduces the number of friendly induced paths, since some paths that did not contain the move may no longer be chordless as a result of the new connections established by the move.

It is likely that shorter induced paths are more valuable than longer ones. The induced path count could be modified to give a higher weight to shorter paths, for instance by introducing an exponential decay factor α. Figure 12.4 shows the number of induced paths through each cell on an empty 10×10 Hex board, with various discount factors. A discount factor of 1.0 corresponds to weighing all paths equally, while a factor of 0.0 considers only the shortest induced paths. The example indicates that a medium setting of 0.5 appears to be more suitable.

Kloks and Kratsch have pointed out that the number of minimal separators can be exponential in $|\mathcal{V}(\mathcal{G})|$. For instance, if the terminals are connected by k independent paths of length l, then any minimal separator consists of a choice of one vertex from each path, for a total of l^k minimal separators. Dual examples can be constructed where the number of induced paths is exponential in the graph size. This would be an apparent problem for induced path based heuristics.

However, the problems of the apparently arbitrary choice of $\alpha = 0.5$ and the exponential number of induced paths can both be addressed by a justifying theoretical observation with an associated efficient approximation algorithm, to be described in the next section.

12.5 Monte Carlo

In Sections 10.6 and 12.4 it was asserted that it is a useful metric to count induced paths weighted by a function of their length, such that the weight of a path of length l is 2^{-l}. There is actually some concrete justification for this method, as it essentially corresponds to a Monte Carlo evaluation metric.[4]

The probability that MAX wins a given set colouring game with random play simply equals the number of colourings that are wins for MAX divided by the total number of colourings. For Hex this corresponds to the number of sets of empty cells that contain a winning path. Let k be the number of empty cells on the board, and let $\mathcal{P}$ be a winning induced path for MAX, with length l. The number of sets of empty cells that contain $\mathcal{P}$, and therefore would be a win for MAX if coloured T, is 2^{k-l}. When comparing two paths, the factor 2^k cancels out and their relative weights have ratios proportional to 2^{-l}. This is why, informally speaking, an induced path that is one cell shorter is twice as good.

The quality of the Monte Carlo evaluation is expected to increase as the move choice within the simulations is improved, since ultimately the correct evaluation would result from optimal move choices in the simulations. Work by Grigoriev in general games [43], and Brügmann [19] and Bouzy and Helmstetter [17] in the game of Go, concentrated on gradually improving the move choice within the simulations by methods such as simulated annealing. A game-specific first step in this direction for Hex is to use only *rational* random moves in the simulations. The metric then corresponds to the fraction of **balanced** sets that are winning sets, where a balanced set contains half the remaining

[4] See Section 11.3.

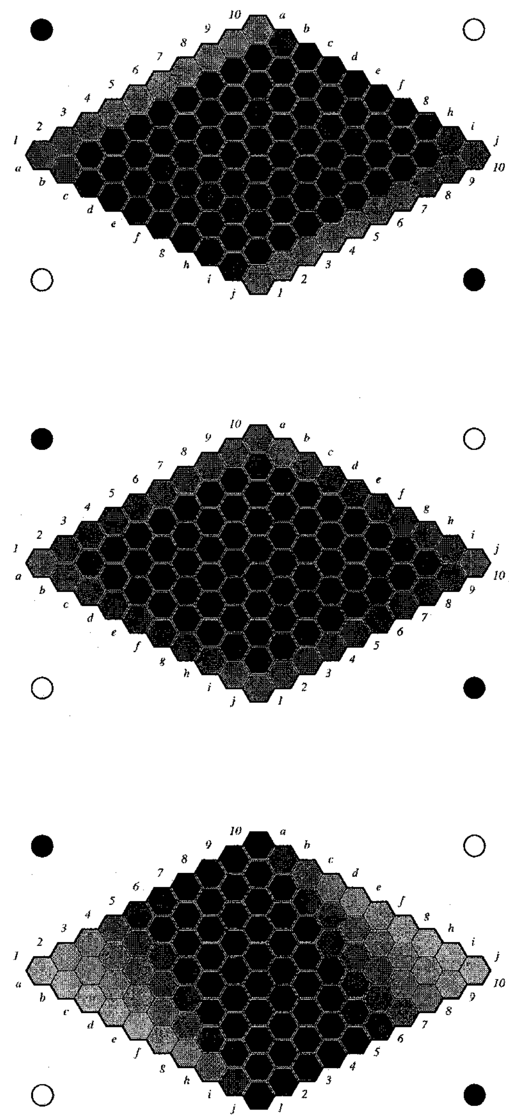


Figure 12.4: Move evaluations for Black based on induced path count with discount factor 1.0 (top), 0.5 (middle), and 0.0 (bottom) in 10×10 Hex. Move evaluation increases with the size and shade of the discs.

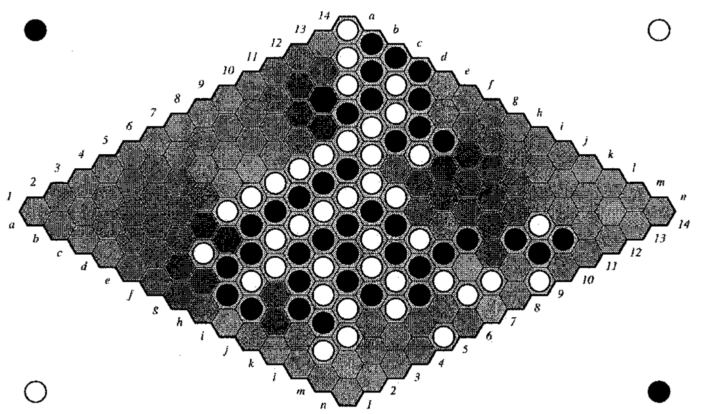


Figure 12.5: Monte Carlo move evaluations for a puzzle by Claude Berge. Darker cells indicate more desirable moves.

empty cells.

Diagram 12.5 shows the rational Monte Carlo analysis of a puzzle by Berge, with White to move. Each move is evaluated as the frequency of wins in games where it was played first by White, which is equivalent to the frequency of wins in games where the move was played by White at any stage of the game, since in set colouring games it does not matter in which order the moves were played. This particular puzzle is further discussed in Section 13.8.

Chapter 13

Dead Cell Analysis

The superrational play criteria set forth in Chapter 8 ultimately depend on the ability to detect dead elements and rational moves. However, as will be shown in Section 13.1, these tasks are NP-complete in general. For the Shannon game fortunately some rules and methods can be stated that will identify most of the important cases. The superrational play criteria then amount to playing locally optimal moves in a game to be called the multi-Shannon game.

13.1 Detecting Live Cells

In the Shannon game, an element will also be referred to as a *node* or *cell*, indicating that the game may be played by colouring vertices in a graph or cells on a game board. Establishing whether a node is live or dead is connected to the following property.

Theorem 13.1.1. Let $\mathcal{G}$ be the game graph of a Shannon game, and let $v \in \mathcal{V}(\mathcal{G})$. Then v is live if and only if there is an induced inter-terminal path that contains v.

Proof.

$\Longleftarrow$: Let $\mathcal{P}$ be an induced inter-terminal path containing v. Consider the colouring $\psi = \text{T}^{\mathcal{P}}\text{F}^{\mathcal{V}(\mathcal{G})\setminus\mathcal{P}}$, which is a complete colouring with a winning path for MAX. The colouring $\psi\text{F}^v = \text{T}^{\mathcal{P}\setminus v}\text{F}^{\mathcal{V}(\mathcal{G})\setminus\mathcal{P}+v}$ does not contain a winning path for MAX, for otherwise $\mathcal{P} \setminus v$ would contain an inter-terminal path and then $\mathcal{P}$ would not be an induced path. Therefore there exists a complete colouring in which the colour of v influences the outcome, and thus v is live.

$\Longrightarrow$: Let v be live, then there exists a complete colouring ψ where the outcomes of ψT^v and F^v differ, which means ψT^v is a win for MAX and ψF^v is a win for MIN because the Shannon game is isotone. Therefore $(\psi\text{T}^v)^{-1}(\text{TRUE}) = \psi^{-1}(\text{TRUE}) + v$ contains an inter-terminal path. Let

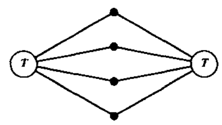


Figure 13.1: No node will be live at the end of the game if both players play rationally.

$\mathcal{S}' \subseteq \psi^{-1}(\text{TRUE}) + v$ be an arbitrary *minimal* set that contains an inter-terminal path. Then $\mathcal{S}'$ is an induced inter-terminal path. We then have $v \in \mathcal{S}'$, for otherwise $\mathcal{S}' \subseteq \psi^{-1}(\text{TRUE}) - v = (\psi \text{F}^v)^{-1}(\text{TRUE})$ and then ψF^v would be a win for MAX. So v is contained in an induced inter-terminal path. □

No nodes are dead in the Shannon game graph for Hex at the start of the game. However, during the course of the game nodes may *become* dead. Figure 13.4 in Section 13.3[1] contained some examples. For any Hex position, Theorem 13.1.1 applies to both the reduced graphs representing the position.

For seasoned game players it may appear as though Figure 13.1 shows a Shannon game graph in which all nodes are on induced inter-terminal paths and yet no complete colouring contains a node whose colour single-handedly determines the outcome, since there will always be two nodes owned by Short. However, recall that in set colouring games the players are not obliged to use their "own" colours, and thus it is legal for the game to end with exactly one node owned by Short. This would of course require Short to have played an irrational move. It would not make sense to modify the definition of dead nodes to consider only "rationally reachable" final positions, since then *all* nodes would be dead from the start in Figure 13.1 which would interfere with the theorems about removing two dead nodes and so on.

Determining whether or not a node is live is therefore equivalent to finding the **monophonic interval** between the terminals, which is the set of all nodes appearing on induced inter-terminal paths. A path between two vertices may also be called **a connector** of the two vertices, and then an induced path is **a minimal connector**. Similarly one may speak of **a separator set** of two vertices, being a set whose removal severs all paths between the two vertices, and the associated concept of **a minimal separator**. For terminal nodes $\mathcal{T} = \{\tau, \tau'\}$ it is obvious that some node v is contained in a minimal τ-τ' connector if and if it is contained in a minimal τ-τ' separator for exactly the same reasons as outlined in the proof of Theorem 13.1.1, since any set $\mathcal{S}'$ is a τ-τ' connector if and only if its complement $\mathcal{V}(\mathcal{G}) \setminus \mathcal{T} \setminus \mathcal{S}'$ is not a τ-τ' separator.

An algorithm due to Kloks and Kratsch finds all minimal separators between two specified vertices in $\mathcal{O}(n^3 R)$ time, where $n = |\mathcal{V}(\mathcal{G})|$ and R is the number of minimal separators [56]. They also point out that R can itself be exponential in n, giving the example of two vertices joined by a set of $\frac{n-2}{2}$ vertex disjoint paths of length 2. Each minimal separator contains one vertex from every path, generating a total number of $2^{(n-2)/2}$ minimal separators.

[1]See page 149.

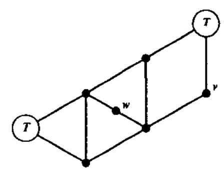


Figure 13.2: Nonlocal influence on life and death: removing node v kills only w.

The problem of determining the monophonic interval between two vertices is connected to the *induced path pairs problem*: Given a graph and a set $\{(v_0, w_0), (v_1, w_1), \ldots, (v_{n-1}, w_{n-1})\}$ of vertex pairs, does there exist an induced subpath consisting of k disjoint induced paths in which for every $i \in \mathbb{Z}_n$ vertex v_i is joined to vertex w_i? A result by Fellows showed that for $k \geqslant 2$ this problem is NP-complete [30]. For $k = 2$ the induced path pairs problem reduces to the problem of finding a monophonic interval by adding a new vertex adjacent to both w_0 and w_1 and asking whether this new vertex is on an induced v_0-v_1 path. Determining membership of a monophonic interval is therefore NP-complete. As a corollary, recognizing dead nodes in the Shannon game is NP-complete [49].

The NP-hardness of detecting dead cells in general is ultimately connected to the fact that the property of life and death is intrinsically nonlocal. Figure 13.2 shows an example where removing one node kills a remote node but none of the adjacent nodes. The graph can be modified to have such "action at a distance" occur arbitrarily far away in the graph.

13.2 Simplicial Nodes

A special class of dead nodes is the one containing all nodes that can be separated from the terminals by *clique cutsets.* A **clique cutset** is a clique whose removal disconnects the graph. For any clique cutset $\mathcal{S}'$ that does not disconnect the terminals, the nodes that are no longer connected to the terminals are evidently dead, since any inter-terminal path containing v must pass through $\mathcal{S}'$ twice and is therefore not chordless. A polynomial-time algorithm by Whitesides finds all clique cutsets in a graph in $\mathcal{O}(nm)$, where $n = |\mathcal{V}(\mathcal{G})|$ and $m = |\mathcal{E}(\mathcal{G})|$ [101]. This was later improved to $\mathcal{O}(n^{2.69})$ by Kratsch and Spinrad [58].

When a clique cutset is found, all nodes not in the same connected component as a terminal are dead. If there are clique cutsets then the algorithms by Whitesides and Kratsch & Spinrad only find one if them. The nodes that are then in the same connected component as a terminal might still be separable from the terminals by another clique cutset. An improvement by Tarjan finds an entire clique cutset decomposition of a graph in $\mathcal{O}(nm + n^2)$ time [98], with a later optimization by Leimer in which all cutsets are minimal and the graph is only decomposed into the irreducible components [63].

The Whitesides and Kratsch & Spinrad algorithms can be modified readily to perform dead node detection. Algorithm 3 is essentially the algorithm presented in [58] with two differences: The

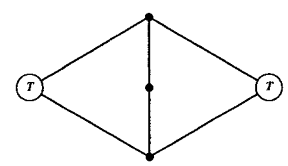


Figure 13.3: The center node is dead, yet there is no clique cutset.

algorithm is initialized with an induced inter-terminal path, and does not halt when a clique cutset is found. These modifications do not change the runtime complexity; in particular, the number of iterations of the outer loop is still $\mathcal{O}(|\mathcal{V}(\mathcal{G})|)$, and the induced path initialization does not exceed the complexity of the outer loop. Thus, Algorithm 3 will determine the clique cutset life or death status of all nodes in $\mathcal{O}(n^{2.69})$ time.

CliqueCutsetLive($\mathcal{G}$, τ, τ')

input : A graph $\mathcal{G}$ with terminal vertices τ, τ'.
output: The set of all vertices not separable from both terminal vertices by clique cutsets.

$\mathcal{P} \leftarrow$ an induced τ-τ' path ;
$\mathcal{S} \leftarrow \mathcal{V}(\mathcal{G}) \setminus \mathcal{P}$;
while $\mathcal{S} \neq \varnothing$ **do**
  $\mathcal{C} \leftarrow$ a connected component of $\mathcal{S}$;
  if $\mathcal{N}(\mathcal{C})$ *is a clique* **then** $\mathcal{S} \leftarrow \mathcal{S} \setminus \mathcal{C}$;
  else
    $v_1, v_2 \in \mathcal{P} \leftarrow$ two nonadjacent neighbours of $\mathcal{C}$;
    $w_1, w_2 \in \mathcal{C} \leftarrow$ neighbours of v_1 and v_2 respectively such that there is a w_1-w_2 path inside $\mathcal{C}$ having no internal vertices adjacent to v_1 or v_2 ;
    $\mathcal{P} \leftarrow \mathcal{P} \cup \{w_1, w_2\}$;
    $\mathcal{S} \leftarrow \mathcal{S} \setminus \{w_1, w_2\}$;
return $\mathcal{S}$;

Algorithm 3: Detecting all clique-cutset-live nodes.

The life and death problem is NP-complete, so the clique cutset algorithm cannot guarantee to establish the correct life or death status of all nodes. It errs on the side of life: Any node that is found to be clique-cutset-dead is indeed dead, but there can be clique-cutset-live nodes that are actually dead as well. Figure 13.3 shows a simple example of a dead node in a graph that has no clique cutsets at all.

A subset of the clique-cutset-dead nodes is formed by all simplicial nodes. Finding all simplicial nodes can be performed by matrix multiplication: If $\mathcal{M}$ is the incidence matrix of a graph, then vertex i is simplicial if $(\mathcal{M}^2)_{ii} = (\mathcal{M}^2)_{ij}$ for all neighbours j of i. Matrix multiplication is also the bottleneck in the Kratsch & Spinrad algorithm for clique cutsets, so finding simplicial nodes does not reduce the time complexity.

The simplicial node condition is intrinsically local. A simplicial node's neighbourhood is a clique cutset separating it from *any* other node, so it is dead independent of the colouring of the rest of the graph, and independent even of the choice of terminal nodes. This allows for partial colourings inducing simplicial nodes in the reduced game graph to be precomputed. The dead cells in Figure 13.4, to be described in the next section, are all derived in this way.

13.3 Basic Hex Patterns

The most basic case of a simplicial node in Hex is based on examining only the direct neighbourhood of the cell on the Hex board. This is not quite the same as simplicial nodes in the reduced Shannon game graph of the position. It is equivalent to uncolouring all cells not adjacent to the target cell, and then checking for simpliciality in the reduced Shannon game graph of the resulting position. Obviously, if a cell is dead in this modified position then it is also dead in the original position, since the original position is a descendant of the modified position and death is by definition irreversible.

This restrictive test for simpliciality will nevertheless ultimately lead to a method that catches almost all dead cells that occur in practice, and, more importantly, all captured and dominated cells. The bottom row in Figure 13.4 lists the five essentially different patterns for dead cells, up to rotation, mirroring, and interchange of colours, based on examining only the immediate neighbours.

From these dead cell patterns one can construct dominated move patterns by removing one piece. The second row from the bottom in Figure 13.4 shows the six essentially different patterns thus obtained. In each case the move in unmarked empty cell produces a pattern on the bottom row, thus killing the empty cell marked '×'. The move in the unmarked empty cell then dominates the move in the marked empty cell for White.

The dominated move patterns are listed again along the vertical axis of the table in Figure 13.4. The entries in the table show combinations of two dominated move patterns, yielding captured sets for White in accordance. Some combinations produce **a reducible captured pattern**, which means that the captured set can be detected using a smaller pattern. In the table in Figure 13.4 the reducible captured patterns are indicated with arrows pointing towards the smaller pattern they contain.

More can be said about the White-dominated patterns in the diagram. Not only should White avoid the marked moves, but Black should avoid them too. The reason is that a Black move in such a cell is reversible by a White move in the dominating cell. Let White be MAX, and let ψ be the colouring of the dominated pattern, in which cell w is dominated by v for MAX. It is then possible to prove by induction that for any position $\mathfrak{p}$ the inequality $\text{MNX}(\Gamma; \mathfrak{p}\psi\text{F}^v) \leqslant \text{MNX}(\Gamma; \mathfrak{p}\psi\text{F}^w)$ holds. To do this, consider the children of both positions. For each pair $\mathfrak{p}\psi\text{F}^v\mathfrak{m}$ and $\mathfrak{p}\psi\text{F}^w\mathfrak{m}$ the inequality holds by induction if $\mathfrak{m}$ is a move outside of $\{v, w\}$. For the moves in v or w two cases are distinguished based on which player is to move. If MAX is to move then the rational moves are $\mathfrak{p}\psi\text{F}^v\text{T}^w$ and $\mathfrak{p}\psi\text{F}^w\text{T}^v$, and we have $\text{MNX}(\Gamma; \mathfrak{p}\psi\text{F}^v\text{T}^w) \leqslant \text{MNX}(\Gamma; \mathfrak{p}\psi\text{T}^v\text{T}^w) = \text{MNX}(\Gamma; \mathfrak{p}\psi\text{T}^v\text{F}^w)$ since T^v kills w. If MIN is to move then the rational moves are $\mathfrak{p}\psi\text{F}^v\text{F}^w$ and $\mathfrak{p}\psi\text{F}^w\text{F}^v$, which lead to the same position. In all cases we have that for each child of $\mathfrak{p}\psi\text{F}^v$ there is a child of $\mathfrak{p}\psi\text{F}^w$ whose minimax value is equal or larger,

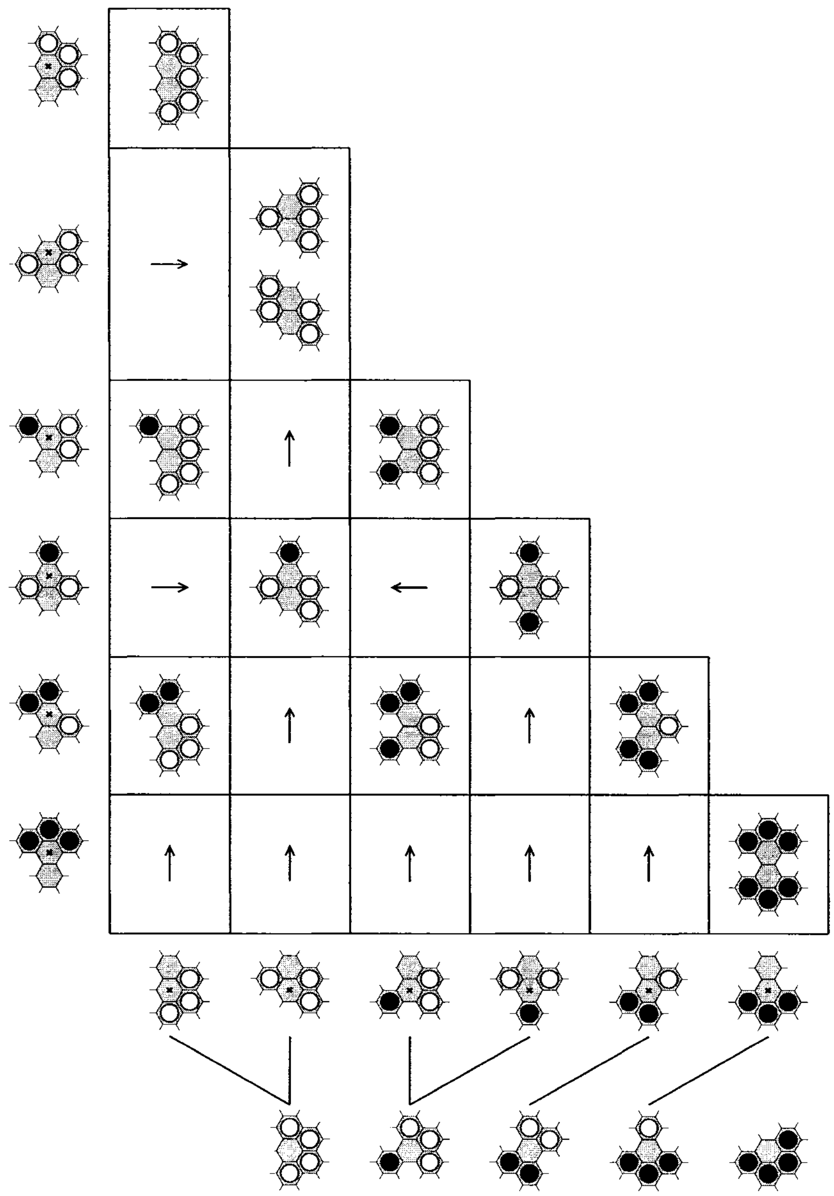


Figure 13.4: Hex patterns for dead cells, and for dominated moves and captured pairs for White.

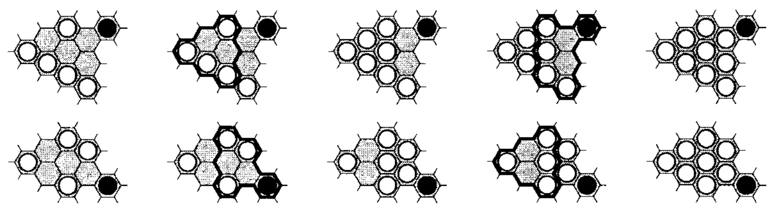


Figure 13.5: Examples of reducible patterns for White-captured sets.

and vice versa. This proves the inequality $\text{MNX}(\Gamma; p\psi F^v) \leqslant \text{MNX}(\Gamma; p\psi F^w)$.

The observations will be subsumed by more general properties of the *multi-Shannon game*, to be described in Section 13.6.

13.4 Reducible Patterns

A computer search for Hex patterns containing captured sets or dominated moves revealed 427 such patterns, up to reflections and rotations. However, many of those patterns are **reducible**, meaning that the information provided by the pattern could also be obtained by applying a smaller pattern. Reducing a pattern into smaller patterns proceeds in two steps: filling in captured cells, and matching smaller dominated move patterns.

The first step, filling in captured cells, is demonstrated in Figure 13.5. The two leftmost patterns are both captured by White. Both of them contain a smaller set that is itself captured by White, independent of the status of the other cells. This smaller set can then be filled in with white pieces. As Figure 13.5 shows, this process can be iterated more than once, as filling in pieces may create new captured patterns. In both examples of Figure 13.5, eventually the whole pattern is filled in with white pieces, confirming that the whole set was indeed captured by White. There are, however, larger captured patterns that cannot be thus reduced; examples will be listed in Section 13.5.

The second step in reducing a pattern is detecting dominated moves using a smaller pattern. In Figure 13.4 this happens to all the table entries containing arrows; the entries are valid patterns, but superfluous, as the dominated moves can also be detected by the smaller pattern pointed to be the arrow. Figure 13.6 shows another example. After fill-in, one of the patterns from Figure 13.4 is detected, and it is concluded that the move indicated by the white dot is preferable for both players to the move with the black dot. This is necessarily true also for the original pattern.

By thus reducing patterns into the basic cases listed in Figure 13.4, almost all of the dead, captured, and dominated moves that occur in Hex practice can be detected. Section 13.8 will list some examples.

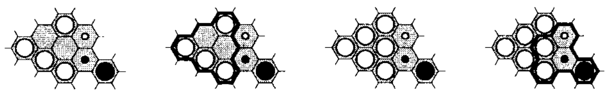


Figure 13.6: Example of a reducible dominated move pattern. The move in the white dotted cell is preferable for both players to the move in the black dotted cell.

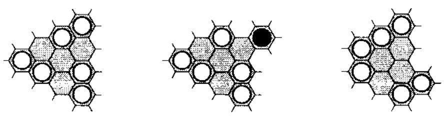


Figure 13.7: Irreducible captured patterns for White.

13.5 Larger Irreducible Patterns

The basic patterns in Figure 13.4 all have at most two empty cells. Larger patterns that are not reducible must also exist. Consider that playing a winning opening move on a board of size $n \times n$ effectively creates an area of size $(n+2) \times (n+2)$, including the border pieces, that must apparently be captured set for the player who played the opening move. Yet the reductions using the basic patterns cannot establish this.

Figure 13.7 lists the three irreducible captured set patterns found by a computer search. Each pattern has the property that no smaller subpattern reveals any captured cells at all. No other irreducible captured set patterns with four connected empty cells were found, nor were any irreducible captured set patterns with three connected empty cells at all.

If some local pattern contains a move that captures the rest of the empty cells, then this move is of course the best possible move within that pattern. If no such move is available, then care must be taken not to play a move $\mathfrak{m}$ that allows the opponent a reply that captures all remaining empty cells, and at the same time kills move $\mathfrak{m}$. A move that would allow such a reply would be the worst possible move to play within the pattern.

Combining these requirements produces Figure 13.8, a list of irreducible patterns with indicated locally optimal moves. A locally optimal move is one that either captures the whole set, or if that is not possible, it prevents the opponent from doing so. In each of the patterns of Figure 13.8, the moves indicated with white markers are moves that capture the entire set for White; they are therefore locally optimal for White. The moves indicated with black markers are moves for Black that stop White from capturing the whole set. If any of these patterns are encountered in a Hex game, only *one* of the marked cells needs to be considered by the appropriate player; all other moves are provably no better.

Note that many of the patterns have more than one locally optimal move for one or both of the players. When there are multiple locally optimal moves, they all dominate each other. This means

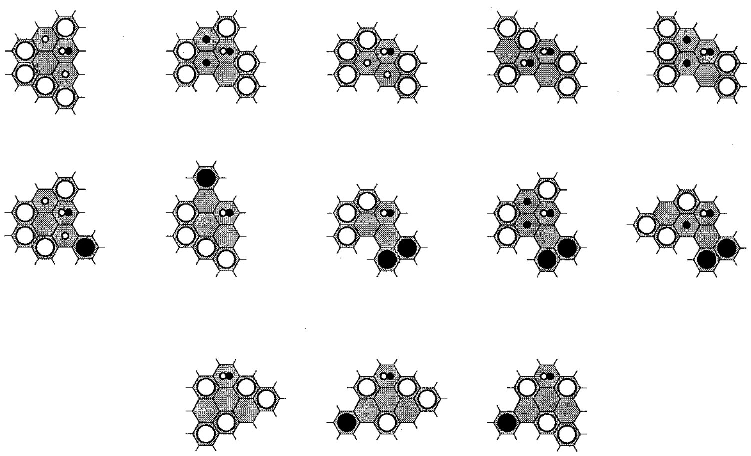


Figure 13.8: Irreducible patterns with locally optimal moves for White and for Black.

that care must be taken not to simply omit any dominated move, for then no move would remain at all. A dominated move may only be omitted provided that at least one of its dominating moves is not. Omitting moves from consideration based on domination is of course transitive: If $\mathfrak{m}$ dominates $\mathfrak{m}'$ and $\mathfrak{m}'$ dominates $\mathfrak{m}''$, then both $\mathfrak{m}'$ and $\mathfrak{m}''$ can be omitted from consideration provided that $\mathfrak{m}$ is not.

In the ten patterns on the top and middle row of Figure 13.8, if Black plays a move that is not indicated with a black marker then White kills this move immediately and captures the whole set. The three patterns on the bottom row do not have this property. If Black makes a mistake, then White will eventually capture the set and kill Black's move, but *not* immediately on the next move.

Any captured set pattern cannot contain an empty cell that has at least two nonadjacent "liberties", where a liberty is an adjacent cell outside the pattern. The reason is that an opponent's first move in such a cell could never be killed from within the pattern anymore: No matter how the rest of the pattern is filled in, the first move could still be part of an induced inter-terminal path that does not touch the rest of the pattern. The capture and domination of sets is closely related to these outside liberties. The local games described in the section on superrational play are in the Shannon game equivalent to a local game based on these outside liberties, namely the *multi-Shannon game.*

13.6 Multi-Shannon

The **multi-Shannon game** is a graph colouring game that can be used to detect superrational moves in a regular Shannon game. It is in fact the graph equivalent of the ternary game based on punctuated formulas as described in Sections 8.6-8.8. As the name suggests, the multi-Shannon game is actually a more general version of the Shannon game itself, and indeed local multi-Shannon games can even be used to find superrational moves in larger multi-Shannon games.

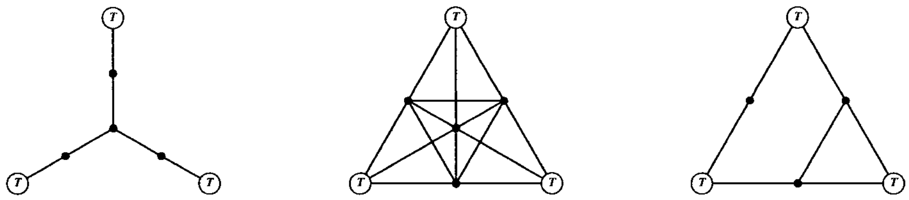


Figure 13.9: Examples of multi-Shannon games: win for Cut (left), win for Short (middle), draw (right).

Definition 13.6.1. Let $\mathcal{G}$ be a graph with a set $\mathcal{T} \subseteq \mathcal{V}(\mathcal{G})$, where the vertices in $\mathcal{T}$ are to be called the *terminal vertices*, or *terminals* for short. The **multi-Shannon game** based on $(\mathcal{G}, \mathcal{T})$ is a game played on the colour space $\mathbb{T}^{\mathcal{V}(\mathcal{G})\setminus\mathcal{T}}$, with the scoring function $f : \mathbb{B}^{\mathcal{V}(\mathcal{G})\setminus\mathcal{T}} \to \mathbb{T}$ defined as

$$\xi \mapsto \begin{cases} +1 & \text{if } \xi^{-1}(\text{TRUE}) \text{ is a } \tau\text{-}\tau' \text{ connector for every connected pair } \tau, \tau' \in \mathcal{T}; \\ -1 & \text{if } \xi^{-1}(\text{TRUE}) \text{ is a } \tau\text{-}\tau' \text{ separator for every nonadjacent pair } \tau, \tau' \in \mathcal{T}; \\ 0 & \text{otherwise.} \end{cases}$$

Some examples of multi-Shannon graphs are shown in Figure 13.9. The game can end in a draw, when some but not all pairs of terminals are connected by Short. It is therefore not strictly a set colouring game according to the definition in Chapter 3. The regular Shannon game is a special case of the multi-Shannon game, played with exactly two terminals. The game of Y is seemingly similar to the multi-Shannon game, having three terminals, but there is a crucial difference. In the game of Y, player Short needs to connect the three sides with *one single chain*. In the multi-Shannon game this is not required. Curiously, this stricter requirement for the game of Y removes the possibility of draws.

The multi-Shannon game can be regarded as a metagame where one regular Shannon game is played between each pair of terminals. All these regular Shannon games are played simultaneously, and the task for both players is to win *all* of them. For player Short this requires connecting every pair of terminals, and for Cut it requires disconnecting every pair. Degenerate cases can arise where some pair of terminals is not even connected at the start of the game, or already adjacent at the start of the game; to deal with this, the players do not need to worry about any terminal pair that *cannot* be connected or disconnected under any colouring.

The multi-Shannon game is actually derived from the games based on achieving optimal colourings in the regular Shannon game. Let $\Gamma = \langle X^{\mathcal{S}}, f \rangle$ be the set colouring game of a Shannon game played on graph $\mathcal{G}$. Consider some arbitrary set $\mathcal{S}' \subseteq \mathcal{S}$ of nonterminal nodes. A maximal colouring of $\mathcal{S}'$ would be a colouring $\psi' \in \mathbb{B}^{\mathcal{S}'}$ with the property that for every complete colouring $\psi^* \in \mathbb{B}^{\mathcal{S}}$ we have $f(\psi^*) = +1 \implies f(\psi^*\psi') = +1$. If ψ^* contains a winning chain $\mathcal{P}$ for Short then:

- Either $\mathcal{P}$ does not intersect $\mathcal{S}'$, in which case ψ' does not destroy $\mathcal{P}$; or
- $\mathcal{P}$ does intersect $\mathcal{S}'$, and since $\mathcal{S}'$ contains no terminals of Γ, $\mathcal{P}$ must therefore also connect two neighbours of $\mathcal{S}'$ with a path inside $\mathcal{S}'$. Then in order for $f(\psi^*\psi') = +1$ there must be a TRUE-coloured chain between these two neighbours in ψ'.

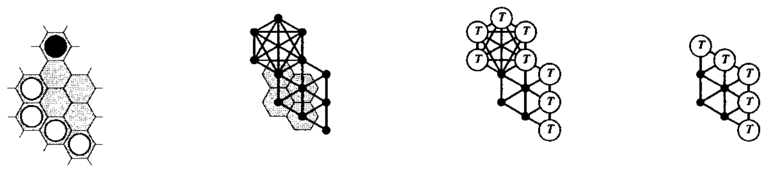


Figure 13.10: Converting a local Hex pattern into a multi-Shannon game, with Black playing Short and White playing Cut.

So to achieve a maximal colouring of $\mathcal{S}'$ it is *sufficient* for Short to connect every pair of neighbours of $\mathcal{S}'$ that have a connecting path inside of $\mathcal{S}'$. It is not always actually *necessary* to do so, because it could happen that there is some pair of neighbours of $\mathcal{S}'$ that never occurs in an induced inter-terminal path in $\mathcal{G}$. When for instance $\mathcal{S}'$ forms a cut set between the terminals in $\mathcal{G}$, then Short really only needs to connect neighbours of $\mathcal{S}'$ that are not in the same connected component of $\mathcal{G}\backslash\mathcal{S}'$.

By very similar reasoning, to achieve a minimal colouring of $\mathcal{S}'$ it is sufficient for Cut to disconnect every pair of neighbours of $\mathcal{S}'$ that are not already adjacent in $\mathcal{G}$. These sufficient requirements for Short and Cut are encoded in the rules of the multi-Shannon game played on $\mathcal{G}(\mathcal{S}' \cup \mathcal{N}(\mathcal{S}'))$ with terminals $\mathcal{N}(\mathcal{S}')$.[2] A winning move in this multi-Shannon game is therefore provably at least as good in Γ as any other move in $\mathcal{S}'$, and any move in $\mathcal{S}'$ is provably at least as good in Γ as a losing move in the local multi-Shannon game.

An example of the process of generating a local multi-Shannon game based on a Hex pattern is displayed in Figure 13.10. The leftmost picture shows the pattern, which is one of the patterns listed in Figure 13.8. The second picture shows a part of the Shannon game graph with Black playing the role of Short. Shown are the nodes corresponding to the empty cells in the pattern, and their neighbourhood. In the third picture, the neighbourhood nodes are turned into terminals. This is already a multi-Shannon game that represents the pattern. However, the multi-Shannon game can be simplified by recognizing that any clique of terminals whose neighbourhoods outside the clique are the same, may as well be replaced by a single terminal. The rightmost picture shows the resulting simplified local multi-Shannon game. If Cut has the first move, then there is only one winning move that guarantees the eventual disconnecting of all separable terminals. Short has no winning first move, but there is one drawing move that guarantees the connection of *some*, but not all, pairs of connectable terminals. This corresponds to the advice given in Figure 13.10.

In general, the local multi-Shannon game induced by $\mathcal{S}'$ has the property that a move that is at least as good in the local game is also at least as good in the global Shannon game. This is true regardless of the choice of set $\mathcal{S}'$, which is a remarkably property; it does not require the global Shannon game to decompose. Following the terminology of Chapter 8, moves that are optimal in a local multi-Shannon game are superrational.

Table 13.6.1 lists the various possibilities for a local multi-Shannon game. If either player has a second-player win, then the set of nonterminal nodes in the multi-Shannon game is captured by that player, and can be filled in. If some player has a first-player win but not a second-player win, then

[2]Recall that the neighbourhood $\mathcal{N}(\mathcal{S}')$ of $\mathcal{S}'$ does not intersect $\mathcal{S}'$ itself.

		Short moves first		
		Short wins	draw	Cut wins
Cut	Short wins	S-captured	–	–
moves	draw	S-dominated	undetermined	–
first	Cut wins	both dominated	C-dominated	C-captured

Table 13.6.1: Possible outcomes of a local game

any locally winning move is a dominating move for that player. The dashes in Table 13.6.1 represent combinations that are impossible, since the multi-Shannon game is isotone. The multi-Shannon game of Figure 13.10 is a first player win for Cut and a draw if Short moves first; it therefore represents a Cut-dominated set.

The procedure to follow is:

1. If there is a local multi-Shannon game with a second-player win for one of the players, the set of nodes is captured and should be filled in. Iterate this step until no more captured sets are found.
2. For any local multi-Shannon game that contains a dominating move, nominate one such dominating move for investigation and ignore the rest.
3. For any local multi-Shannon game that contains a dominating move for the opponent, ignore all the locally losing moves.

All the actions of this procedure fulfill conditions that are sufficient for superrational play. As stated before, the conditions may not always be necessary; in terms of local multi-Shannon games this means that a move that is a local multi-Shannon draw may still be superrational for Short or Cut.

As a final note, it can be remarked that the described procedure is also superrational when the local multi-Shannon game occurs within a larger multi-Shannon game; the argumentation is precisely the same.

13.7 Efficient Pattern Matching in Hex

In order to detect patterns occurring around a given cell efficiently, the following encoding can be used. The board around the given cell is divided into six "slices". The cells in each slice are numbered as shown in Figure 13.11. These numbers are used as bit positions in a binary encoding of the contents of the slice. This way, any pattern within a radius of at most seven cells can be specified by six integers. However, patterns need to specify four possible cell states: empty, black, white, or irrelevant. This could then be done by using two binary numbers per slice; Table 13.7.1 specifies the encoding.

Figure 13.12 shows an example featuring the first pattern occurring in Figure 13.5. The two numbers

bit in first number	bit in second number	encoding
0	0	cell state is irrelevant
0	1	cell is empty
1	0	cell contains a black piece
1	1	cell contains a white piece

Table 13.7.1: Bit encoding for patterns.

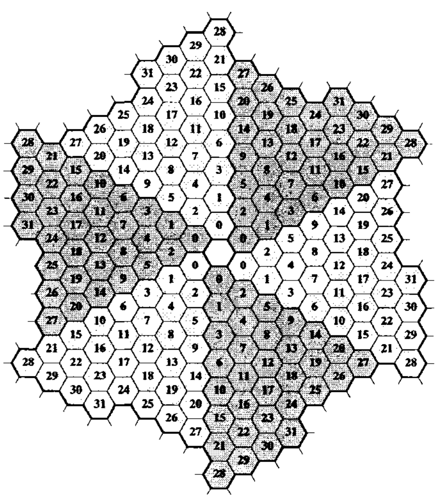


Figure 13.11: Numbering of cells for Hex pattern encoding.

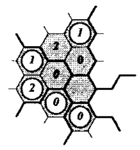


Figure 13.12: Example pattern encoding.

rotation number	ring pattern	first number	second number	hash location
0	e,e,w,w,i,i	001100	001111	783
1	e,w,w,i,i,e	011000	011110	1566
2	w,w,i,i,e,e	110000	111100	3132
3	w,i,i,e,e,w	100001	111001	2169
4	i,i,e,e,w,w	000011	110011	243
5	i,e,e,w,w,i	000110	100111	423

Table 13.7.2: Bit encoding for patterns; e = empty cell, w = white piece, i = irrelevant.

for the top slice are ...00010 and ...00111 in binary, or 2 and 7 in decimal. The other slices, going counter-clockwise, are: (6, 7); (1, 1); (1, 1); (0, 0); (0, 0). Apart from detection speed, the other considerable advantage of this encoding is that rotating a pattern by 60 degrees is trivial. Mirroring a pattern is not trivial, so the best thing to do is to specify asymmetrical patterns twice, once for each handedness.

To avoid scanning the entire table of patterns for possible matches, each pattern can be hashed to one twelve-bit number based on the "ring pattern", defined as the state of the six cells surrounding the center cell. For the pattern of Figure 13.12 the binary numbers for the ring pattern are 001100 and 001111, again going counterclockwise starting with the top cell as the least significant bit. This leads to the binary number 001100001111, or 783 in decimal. The hash table will then store a reference at location 783 to the pattern, along with the rotation. Each pattern thus has six references, along with rotation numbers, stored in the hash table. The six references for the pattern of Figure 13.12 are listed in Table 13.7.2.

To scan for possible matches surrounding a given cell on the board, its ring number is calculated and only the patterns stored at the corresponding hash location need to be checked. The slice numbers surrounding the given cell are calculated and matched against the slice numbers of the stored pattern, in the orientation that was stored at the hash location. In the ideal case this reduces the number of patterns to be checked for possible matches by a factor of approximately $6 \times 4^6 = 24576$ on average, though in practice there will be some clumping as the four possible cell states do not occur with equal frequency in the six ring cells. For further efficiency, note that the slice numbers and ring numbers of any given cell can all be calculated incrementally, as each move on the board changes at most one bit in the slice numbers and the ring number.

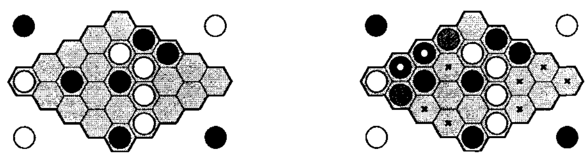


Figure 13.13: Berge's puzzle 1 with White to move and win (left), and its status after dead cell analysis.

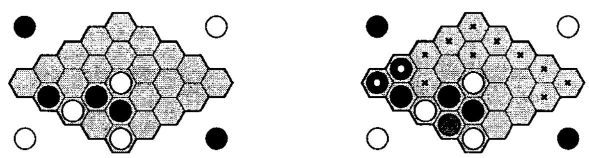


Figure 13.14: Berge's puzzle 2 with White to move and win (left), and its status after dead cell analysis.

13.8 Examples

Figures 13.13–13.17 show the results for the puzzles published by Berge in [11]. Grey pieces marked with a cross indicate dead cells; marked black and white pieces indicate captured cells, and empty cells marked '×' indicate dominated moves for the player to move next.

Figures 13.15 and 13.16 show the importance of iterating the fill-in of captured sets. Both diagrams show a long "corridor" entirely filled in with captured pieces by successively detecting two more captured pieces after each fill-in. Figure 13.17 shows how the battle is effectively already decomposed into two unconnected areas, even though in the original position those two areas are still connected.

Berge's puzzle 5

The following discussion of Berge's puzzle 5 will serve as a more elaborate examples of the application of dynamic traces and dead cell analysis. Due to the large board size and complicated strategic nature of the puzzle, the correct analysis was not previously known.

Berge's puzzle 5 with White to move is a win for White. Figure 13.18 shows a minimal dynamic trace. White starts with 1. c11 and threatens to connect c11 to the white string at e5-e12, which has a virtual connection to the southwest. If Black merely concentrates on stopping White from linking up these two chains, play continues with 1. ... d10; 2. c10 d9; 3. c9 d8; 4. c8 d7; 5. c7 d6; 6. c6 d5; 7. c5. The most challenging continuation by Black is then 7. ... d3, followed by 8. d2! and White connects via b4 or d4.

Black can complicate matters by threatening White's virtual connection of the e5-e12 string. If

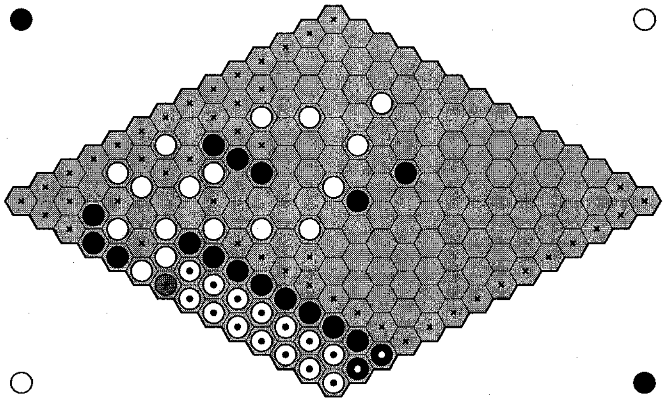


Figure 13.15: Dead cell analysis of Berge's puzzle 3 with Black to move and win.

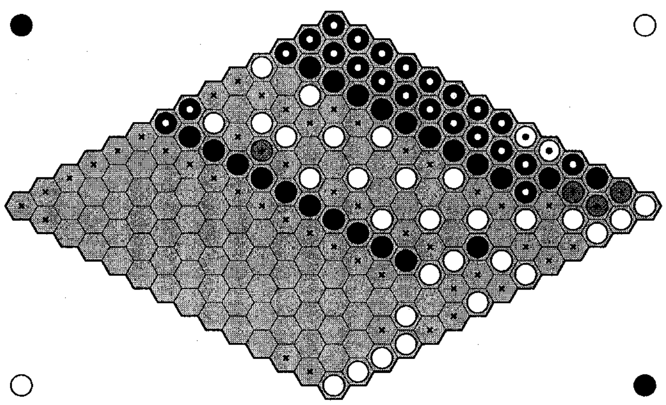


Figure 13.16: Dead cell analysis of Berge's puzzle 4 with Black to move and win.

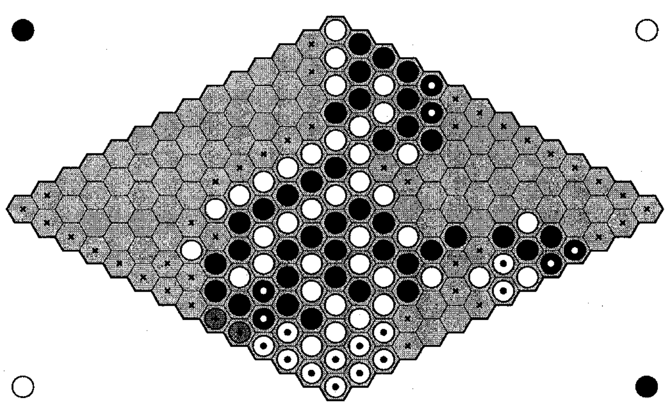


Figure 13.17: Dead cell analysis of Berge's puzzle 5 with White to move and win.

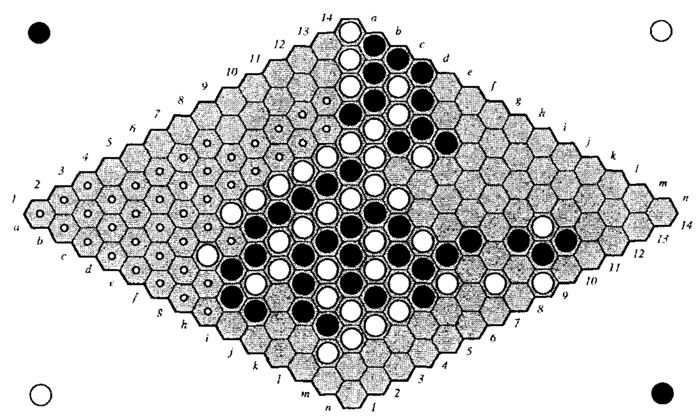


Figure 13.18: Dynamic trace for White's win in Berge's puzzle 5.

Black plays e4 early on, then White does *not* respond with repairing the connection at f4, but with the stronger move d5. This not only repairs the connection to the southwest – though in a less direct manner – but at the same time establishes a virtual connection with the c11-c12 string.

If Black inserts the move e2, then White does play a simple repair move in the f1-f3-h1 triangle. The main line then continues as before until 7. c6 d5; 8. c5, and then for example 8. ... d5; 9. c4 d1; 10. d3 e3; 11. d2 e1; 12. b2.

The most difficult challenge that Black can mount is the following variation: 1. ... g1; 2. d3 e2; 3. c4. Both moves by White are the unique winning replies, ignoring useless delaying moves in Black-captured sets, and are very hard to find for human players. Of interest is the continuation 3. ... d5; 4. c7 e3; 5. b6, which is the reason that the cell b7 is in the minimal dynamic trace. These two winning moves by White are not unique, but other winning moves only serve to delay matters until the moves in question are required anyway.

The opening move 1. c10 loses for White: 1. ... c11; 2. a12 b11; 3. a11 b10; 4. a10 b9; 5. a9 b8; 6. a8 b7; 7. a7 b6; 8. a6 c4 and then for example 9. b4 c2; 10. c3 d2; 11. d3 e2; 12. f2 e3; 13. d4 e4; 14. f4 d5, or 9. a4 b2!!; 10. b3 c2 and so on, or 9. a4 b2; 10. b5 c5. The opening move 1. b11 loses too, 1. ... c11 2. b12 c10 3. b10 c9 4. b9 c8 5. b8 c7 6. b7 c6 7. b6...

In Berge's discussion of puzzle 5 he concentrates on White's attempt to connect the piece at g11 to the northeast. This is evidently not necessary, as White can connect a14 to the southwest, but to answer Berge's question the puzzle can be modified by flipping the colour of the piece at a14. In the resulting position it would be sufficient for White to connect g11 to the northeast. However, Figure 13.19 shows a dynamic trace for *Black*, which means that White cannot connect to the northeast at all.

The important concept to notice in Figure 13.19 is that Black's k9 group has a *triple* connection to the southeast, meaning that after any White move the group still has a virtual connection to

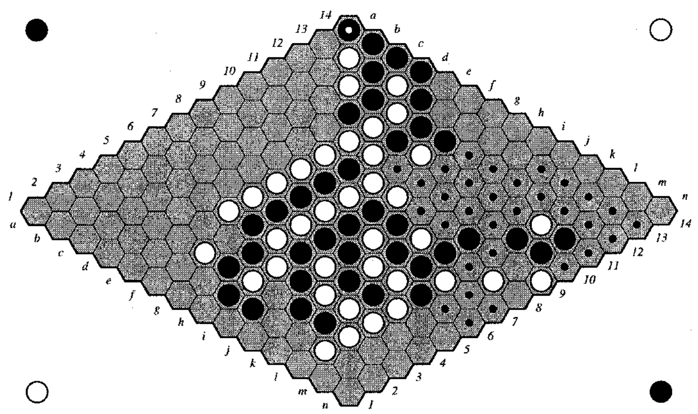


Figure 13.19: Dynamic trace for Black in modified version of Berge's puzzle 5.

the southeast. Black's threat in Figure 13.19 is to play i11, and then i10 or j10 on the next move. The i11 group would then have a virtual connection to the a14-g12 group. It would have a triple connection to the k9 group, which in turn has a triple connection to the southeast. This implies that is a virtual connection no matter what White's intermediate move was.

The only first moves by White that prevent this scenario are i11 and h11. The move 1. i11 loses immediately to 1. ... h10 and Black has a virtual connection from the a14-f11-g12 group to h10 and from h10 to k9. The remaining move to investigate is the main line, where White plays 1. h11. Black responds 1. ... h12 and threatens to win at i11 or j11. Ignoring the Black-captured cells n10 and n11, White's only option is to play 2. i11. This is followed by 2. ... i12. Black then wins at j11 on the next move, for instance 3. k11 j11; 4. k10 j10; l9 m7, except when White preempts by playing 3. j11 after which Black connects with 3. ... j12; 4. k11 k13.

Berge's discussion centers on the main line 1. k10 l9; 2. h11. This indeed wins for White, but unfortunately Black's move at l9 was actually a blunder. Berge mentions the alternative reply 1. ... k11, which in reality does win for Black, but incorrectly concludes that White has a winning reply in this variant too. In later publications extra white pieces appear at m7 and n5, and in that case Berge's analysis does hold. However, the two extra white pieces disturb the "study" nature of the position; where Figure 13.18 was in all likelihood taken from an actual game, adding white pieces at m7 and n5 creates a position that cannot arise in a legal game as there is an imbalance in the number of black and white pieces.

Chapter 14

Discussion

To conclude this thesis, computational results and open questions follow below, with suggestions for further research and experimentation.

14.1 Hex Opening Positions

Using, among other methods, the notions of "win patterns", "domination", and "fill-in", the 7×7 opening position for Hex was solved [46, 47]. A small opening book of fully solved 7×7 positions and an extensive book for 6×6 positions has been calculated by the *Queenbee* program, and is available online [81]. These openings were previously intractable computationally [83]. Figures 14.1 and 14.2 show some results

The goal of solving the 8×8 opening position has not yet been achieved. The shape of the winning region on the empty 8×8 board is a matter of great curiosity, considering that the winning regions on 2×2 through 6×6 become progressively less complicated but the winning region on the empty 7×7 is peculiarly complex. For solving 8×8 Hex, a large parallel computation using the methods

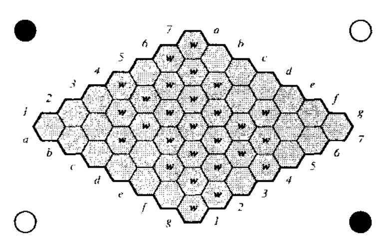


Figure 14.1: Winning moves, marked *'w'*, for White in 7×7 Hex.

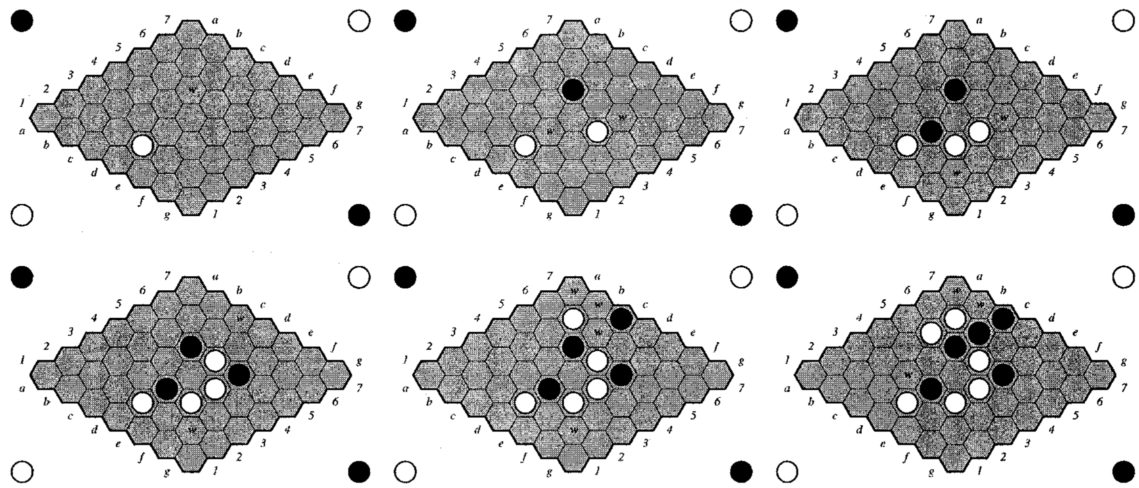


Figure 14.2: Winning moves, marked *'w'*, for Black in a main line after White's opening move d2.

described in this thesis combined with the standard opening book generating algorithm by Buro [20] will likely be required.

It may additionally be useful to pre-generate all possible induced paths on the 8×8 board, since the number of these paths is a mere 2,195,830.[1] A possible approach is to pre-generate only the induced paths up to a given length, which imposes a stricter winning condition on the player who has the first move. An iterative deepening algorithm that adds incrementally longer paths at each iteration may be very efficient. The number of induced paths of any given length on the empty 8×8 board is listed in Table 14.1.1. It can be seen that the number is very manageable for the early iterations.

14.2 Hex Playing Strength

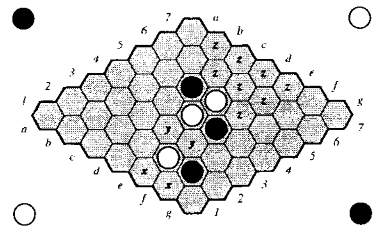
For heuristic play on larger board sizes, such as the commonly used 10×10 or 11×11, the state of the art as of 2005 is the program *Six*, based on virtual connections. The following approaches have not yet been tested fully:

- Random game tree search or playout analysis, Section 11.1;
- Static analysis of board states using Monte Carlo sampling, Section 12.5;
- Lambda search, Section 11.1;
- Decomposition dynamic trace search, to be described below.

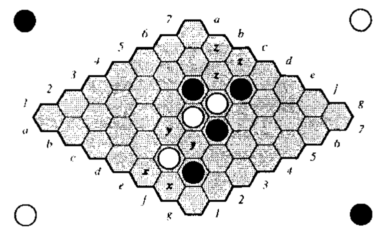
[1] See Section 10.6.

length	paths	cumulative
8	576	576
9	1,602	2,178
10	3,087	5,265
11	4,854	10,119
12	8,801	18,920
13	15,558	34,478
14	28,694	63,172
15	49,148	112,320
16	80,013	192,333
17	116,054	308,387
18	157,291	465,678
19	204,192	669,870
20	253,332	923,202
21	290,992	1,214,194
22	298,526	1,512,720
23	263,852	1,776,572
24	197,199	1,973,771
25	127,108	2,100,879
26	63,866	2,164,745
27	23,376	2,188,121
28	6,306	2,194,427
29	1,288	2,195,715
30	115	2,195,830

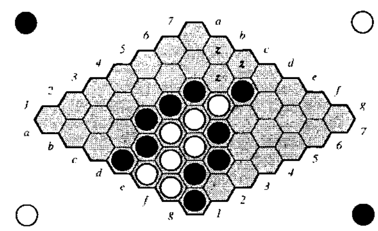
Table 14.1.1: Number of induced paths of given length on the empty 8×8 Hex board.



I. Black to move loses because of three independent local patterns.



II. Black tries one reply...



III. White conjectures away the seemingly unrelated parts of the pattern.

Figure 14.3: Automatically decomposing a dynamic trace for Hex.

The first two methods are essentially the same, as Monte Carlo sampling of a board state is identical to random search with a branching factor of 1.

The dynamic trace search algorithm described in Section 11.2 finds only global connections and is not able to decompose connections into independent sub-parts that can be verified independently. A possible method for achieving the decomposition was described in [84] and reproduced here.

The dynamic trace in Figure 14.3-I contains three independent local connections. Whenever Black plays in one of these three connections, White replies in the same one. Each of the local connections could be proved independently. Dynamic trace search does not recognize this; for each move that Black tries in region z, the subsequent search proves the connections at x and y again from scratch.

When Black tries the move in Figure 14.3-II, the search returns a win for White with the indicated dynamic trace. Black now notices that this pattern consists of three groups of cells. Two of those groups were not touched by the latest black move played. So Black may now conjecture the following:

If the position is indeed a loss, *and* the losing dynamic trace consists of independent sub-patterns, *then* the latest move played only interfered with one of those sub-patterns.

Let the **interfered groups** be those groups of the pattern in 14.3-II that are adjacent to Black's last move, and call the other groups **untouched**. If the conjecture is true, then the untouched groups are strong local connections, meaning the connection can be established even when the defender moves first, and the interfered group is a weak local connection, which means the connection can be established if the attacker moves first. The weak local connection would then be part of a strong local connection that is not yet fully discovered, namely connection z in Figure 14.3-I.

To prove the conjecture, Black can alter the position as shown in 14.3-III. The untouched sub-patterns are replaced by white pieces, to solidify their connections, and they are surrounded by black pieces. The latter is necessary, because the conjecture is that the remaining parts of the losing pattern in 14.3-I are independent from the part that is yet to be discovered. It must therefore be enforced that White wins without using any of the cells adjacent to the untouched groups, which is achieved by adding the surrounding black pieces.

In the position thus created, Black needs only to check the cells in the interfered group of 14.3-II. These are the three cells indicated in 14.3-III. *If* this yields a loss for Black, *then* position 14.3-I is also a loss for Black, and the threat pattern for 14.3-I is the one in 14.3-III plus the untouched groups of 14.3-II.

14.3 Open Questions

GAME-SAT versus QBF

Neither QBF nor GAME-SAT are more expressive than SAT, but they are in many cases more economical in their representation, sometimes exponentially so. When comparing QBF to GAME-SAT, it may be that for many "realistic" games the GAME-SAT representation is more economical in practice. The standard QBF representation with alternating existential and universal quantifiers imposes the order in which the elements are to be coloured, whereas GAME-SAT leaves the players free in this choice. It may be worth investigating how the two paradigms compare in their economy of representation of many games. The comparison would be unfair for set colouring games themselves, as GAME-SAT *is* the set colouring game for two colours, but the comparison may be relevant for different types of games.

The superrational play strategy applies equally well to QBF and SAT solvers, as it is based on optimal colourings which occur in QBF and SAT as well. Indeed as remarked in Section 11.4, the commonly used heuristic of pure literals is a special case of a superrational move. It would be worthwhile to investigate how often an optimal colouring occurs in benchmark QBF and SAT instances, both for random instances and for instances based on "real" problems.

Limitations of Set Colouring Games

In Section 4.8 it was reiterated that set colouring games do not have any restrictions on choice of element or colour. It seems unlikely that there is a way to map a game with such restrictions to an equivalent set colouring game, but a definite proof is still lacking.

A game with restricted colour choice can under some circumstances be modelled as a set colouring game. Hex itself is an example thereof; if the game is isotone, then it does not matter if the players are restricted to using only their "own" colour. This is not the case when players are restricted to using only their opponent's colour, as in Reverse Hex. The trivial case of 1×1 Reverse Hex is the game called $\mathrm{Z}_{\triangle}$ in Chapter 7, which does not have an equivalent as a set colouring game of the same dimension.

However, it does in some sense have an equivalent as a set colouring game played on two variables, namely $\mathrm{Z}_{\square}$. In some cases there might be a mapping from a winning strategy in some game to a winning strategy in another game of a different dimension. Is this possible?

For games with restricted choice of element to colour, perhaps in some cases it is possible to construct an equivalent set colouring game, particularly when a player's right to move in a certain element never changes during the game as long as the element is uncoloured. Then each player has a static subset of elements in which to move. Perhaps it is possible to augment such a game with "gadgets" that will dissuade a player form ever moving in a certain element. Something akin to this happens for instance in GAME-SAT when played on the formula $(t_0 \equiv t_1) \wedge \ldots$, in which moving in t_0 or t_1 guarantees immediate defeat for MAX, nor guarantees neither immediate defeat nor immediate victory for MIN.

Combinatorial Game Theory

In this exploration of a new theory of binary combinatorial games, the following open questions and conjectures have suggested themselves.

- Are there division games or GAME-SAT instances with canonical forms different from the ones encountered so far? New forms could only arise from non-decomposable games. Computer searches have found any new forms.
- If there are other canonical forms, is it still true that each canonical form only occurs in one particular parity?
- If there are other canonical forms, do they obey the "$G \geqslant H$ if no $G^R \leqslant H$ and $G \leqslant$ no H^L" rule?
- Is there a recursive rule for the $\geqslant$ relation that does not invoke combinatorial values?
- Is it strange that there can be binary games that are "truer than true", for instance $\mathrm{T} \uparrow > \mathrm{T}$?
- Are there strategic rules to explain the remaining cases of metagames in Table 7.8.1?

- What does it mean that the candidate values +1 and −1, if they do occur, are not identical to T and F? Is it safe to simply choose T = +1 and claim that the values are not new?
- Does the third candidate value, the off-parity zero, occur in division games or GAME-SAT?

Conjectures:

- Never take a star, unless the game is $\mathrm{Z}_{\triangle}*$ or $\mathrm{Z}_{\square}*$.
- For the assertion $G \geqslant H$ in division games, if $G \vee \overline{H}$ is even then the condition $G \vee \overline{H} > 0 > \overline{G} \wedge H$ is not only necessary but also sufficient; if $G \wedge \overline{H}$ is odd then the condition $G \wedge \overline{H} \geqslant 0 \geqslant \overline{G} \vee H$ is not only sufficient but also necessary.

These conjectures are formulated by checking the sixteen known canonical forms, for which they do hold.

Shannon Game Graphs

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and $\mathcal{G}' = (\mathcal{V}, \mathcal{E}')$ be two Shannon game graphs. Then the graphs are each other's **Shannon dual** if any set $\mathcal{S}' \subseteq \mathcal{V}$ is a terminal connector in $\mathcal{G}$ if and only if it is a terminal separator in $\mathcal{G}'$. Playing the Shannon game on $\mathcal{G}$ is equivalent to playing on $\mathcal{G}'$ with the roles of Short and Cut interchanged. The graphs may have different edge sets, but the vertices must be the same, as they correspond to the game moves and the two terminals.

Such pairs of graphs are indeed dual, since obviously any graph is the Shannon dual of its own Shannon dual, if it has one. Hex graphs have a Shannon dual, as shown in Figures 4.5 and 10.4. But it seems unlikely that every graph has a Shannon dual. It is an open question what the necessary and sufficient requirements are for the existence of a Shannon dual. Planarity is not required, as the graphs in Figure 10.4 are not planar.

This is related to the more general question of which coalition games can be represented by a Shannon game graph. Hex is an isotone game and therefore a coalition game, and every coalition game has dual representations as a DNF and a CNF. But it is certainly not true that every CNF or DNF can be represented with a Shannon game graph. For instance, the game of Y is also isotone, but the size-2 Y board does not have a Shannon game graph.

14.4 Conclusion

The following quote is commonly ascribed to chess player Edward Lasker[2] speaking of the game of Go:

[2]Not to be confused with former chess world champion Emmanuel Lasker.

> Chess is a game restricted to this world, Go has something extraterrestrial. If ever we find an extraterrestrial civilization that plays a game that we also play, it will be Go, without any doubt. – Edward Lasker.

This notwithstanding the fact that Go does have some idiosyncrasies, evidenced by the existence of various rule sets and special cases for game outcomes. To conclude this thesis I would like to put forward a stronger assertion:

> Hex has a Platonic existence, independent of human thought. If ever we find an extraterrestrial civilization at all, they will know Hex, without any doubt.

Hopefully, they will have solved 8×8 Hex.

Appendix A

Appendices

A.1 Notation Conventions

Table A.1.1 contains a list of constants. Throughout the text, fixed variable naming conventions have been used. These conventions are tabulated in Table A.1.2, with references to their type definitions.

A.2 Operators and Functions

The relations and functions between the various variable types, as defined in the text, are listed in Tables A.2-A.2. Table A.2 contains the functions with just one variable type, including unitary functions. Tables A.2 and A.2 display the functions of two or more variable types. There are three functions, namely $\langle X^{\mathcal{S}}, t\rangle$, $\langle X^{\mathcal{S}}, f\rangle$, and $f(\mathfrak{p}, \mathfrak{t})$, that take three variable types. These are listed for each of the three combinations of two of their input types.

A.3 Sam Lloyd's Comet

The "Comet" chess problem mentioned by Berge, now mostly lost in history, is reproduced in Figure A.1.[1] The solution has the white king travelling to g5, while Black can do nothing but shuttle the bishop between h2 and g1. White then delivers mate with Kxh4 and then Rxg3. The move Rxg3 needs to be timed such that the black bishop is on g1.

However, during this maneuver the white king cannot enter a white square, lest black checks the

[1] Chess diagram created courtesy of Dirk Bächle's `fen2eps` software.

symbol	explanation	see definition
$\varnothing$	the empty set	Section 2.1
$\mathbb{N}$	the set of integers including 0	Section 2.1
$\mathbb{Z}_n$	the set of integers $\{0, 1, \ldots, n-1\}$	Section 2.1
$\mathbb{B}$	the set of Boolean values	Section 2.1
FALSE, TRUE	members of $\mathbb{B}$	Section 2.1
$\mathbb{T}$	the set of ternary values	Section 2.1
ϕ	the ternary value "unknown" or "unassigned"	Section 2.1
$\mathbb{X}_n$	$n \times n$ Hex board	Section 2.6
$\mathbb{Y}_n$	size-n Y board	Section 4.6
$\mathfrak{C}$	the set of players	Definition 3.1.2
MIN, MAX	the two players	Definition 3.1.2
λ	bijection from $\mathfrak{C}$ to $\mathbb{B}$	Definition 3.1.2
T, F	stopping positions of a binary combinatorial game	Definition 7.1.1
X, $\mathrm{Z}_\triangle$	combinatorial values for a simple switch and odd zero	Section 7.6
$\mathrm{Z}_\square$	combinatorial value for even zero	Section 7.7

Table A.1.1: Constants.

symbol	explanation	see definition
i, j	an integer	
k	an integer, usually the dimension or cardinality of a set	
$\mathcal{S}$	a subset of $\mathbb{N}$	
v, w	a set element or graph node	
t	a truth value, element of $\mathbb{B}$ or $\mathbb{T}$	
$\mathcal{F}$	a family of subsets of $\mathbb{N}$	
X	a set of colours	Definition 2.1.1
χ	a colour	Definition 2.1.1
ψ	a colouring	Definition 2.1.1
f	a scoring function, usually $X^{\mathcal{S}} \to \mathbb{B}$	
$\mathcal{G}$	a graph	Section 2.5
$\mathcal{P}$	a path in a graph	Section 2.5
Γ	a game	Section 3.1
$\mathfrak{c}$	a player, element of $\mathfrak{C}$	Definition 3.1.2
$\mathfrak{m}$	a move	Definition 3.2.2
$\mathfrak{p}$	a position	Definition 3.2.3
$\mathfrak{t}$	a transition	Definition 3.2.5
$\mathfrak{s}$	a strategy	Definition 3.3.1
h	a pseudo-homomorphism	Definition 3.4.1
$\mathcal{Q}$	a family of games	Definition 6.2.1
G, H, K	a binary combinatorial game	Section 7.1

Table A.1.2: Naming conventions for variables and functions.

	unary		binary	
$\mathcal{S}$	$\mathcal{K}_{\mathcal{S}}$	[S2.5]		
	$\mathfrak{P}_{\mathcal{S}}$	[D3.2.3]		
	$\mathfrak{S}_{\mathcal{S}}$	[D3.3.1]		
v, w	$\mathcal{N}(v)$	[S2.5]	$v \sim w$	[S2.5]
t	$\langle \varnothing, t \rangle$	[D3.1.4]	$t \equiv t'$	[S2.3]
X	$\mathring{X}$	[S2.1]		
ψ	$\mathcal{D}(\psi)$	[D2.1.1]	$\psi \subseteq \psi'$	[D2.1.1]
	$\mathcal{U}(\psi)$	[D3.2.1]	$\psi\psi'$	[D2.2.1]
	$\mathcal{A}(\psi)$	[D3.2.1]	$\psi \longrightarrow \psi'$	[D2.2.3]
	$\overline{\psi}$	[D3.2.1]	$\psi \succ \psi'$	[D2.2.4]
	$\mathfrak{M}(\psi)$	[D3.2.2]	$\psi \succcurlyeq \psi'$	[D2.2.4]
			$\psi \trianglerighteq \psi'$	[D2.2.4]
$\mathcal{G}$	$\mathcal{V}(\mathcal{G})$	[S2.5]	$\mathcal{G} \subseteq \mathcal{G}'$	[S2.5]
	$\mathcal{E}(\mathcal{G})$	[S2.5]	$\mathcal{G}_1 \cup \mathcal{G}_2$	[S2.5]
$\mathfrak{c}$	$\overline{\mathfrak{c}}$	[D3.1.2]		
$\mathfrak{p}$	$\mathcal{U}(\mathfrak{p})$	[D3.2.3]	$\mathfrak{p} \longrightarrow \mathfrak{p}'$	[D3.2.4]
	$\mathfrak{M}(\mathfrak{p})$	[D3.2.3]		
$\mathfrak{s}$			$\mathfrak{s} * \mathfrak{s}'$	[D3.3.3]
$\mathcal{Q}$			$\langle\langle \mathcal{Q}, \wedge \rangle\rangle$	[D6.2.3]
			$\langle\langle \mathcal{Q}, \vee \rangle\rangle$	[D6.2.3]
Γ	$\Gamma *$	[D3.1.5]	$\Gamma \wedge \Gamma'$	[D6.2.3]
	$\Gamma * \Box$	[D6.1.6]	$\Gamma \vee \Gamma'$	[D6.2.3]
	$\Gamma * \triangle$	[D6.1.6]	$\Gamma \geqslant \Gamma'$	[D6.3.1]
G, H, K	$\overline{G}$	[S7.1]	$G > H$	[S7.2]
	$G*$	[S7.5]	$G = H$	[S7.2]
	$G * \Box$	[S7.5.2]	$G \| H$	[S7.2]
	$G * \triangle$	[S7.5.2]	$G \geqslant H$	[S7.3.1]
			$G \mathrel{\vert\!\rhd} H$	[S7.3.1]
			$G \wedge H$	[S7.4.1]
			$G \vee H$	[S7.4.1]

Table A.2.1: Functions and relations with one variable type; numbers in brackets refer to the Section of Definition number

	$\mathcal{S}$	v, w	t	$\mathcal{F}$	X
t	$\langle X^{\mathcal{S}}, t\rangle$ [D3.1.4]				
X	$X^{\mathcal{S}}$ [D2.1.2] $\langle X^{\mathcal{S}}, f\rangle$ [D3.1.3] $\langle X^{\mathcal{S}}, t\rangle$ [D3.1.4]		$\langle X^{\mathcal{S}}, t\rangle$ [D3.1.4]	$\mathrm{COAL}(\mathcal{F}; X)$ [D2.4.3]	
χ	$\chi^{\mathcal{S}}$ [D2.1.1]	χ^{v} [D2.2.1]			
ψ	$\mathrm{PROJ}_{\mathcal{S}\to\mathcal{S}'}(\psi)$ [D2.1.3] $\psi \searrow \mathcal{S}$ [D2.1.3]	$\psi(v)$ [D2.1.1] ψ_v [D2.1.1]			
f	$\langle X^{\mathcal{S}}, f\rangle$ [D3.1.3]				$\langle X^{\mathcal{S}}, f\rangle$ [D3.1.3]
$\mathcal{G}$	$\mathcal{G}\backslash\mathcal{S}$ [S2.5] $\mathcal{G}/\mathcal{S}$ [S2.5]	$\mathcal{N}_{\mathcal{G}}(v)$ [S2.5] $v \overset{\mathcal{G}}{\sim} w$ [S2.5]			
$\mathfrak{p}$	$\mathfrak{p} \searrow \mathcal{S}$ [D3.2.3]				
Γ	$\Gamma * \mathcal{S}$ [D6.1.5] $\Gamma^{+}_{\mathcal{S}}$ [D8.2.2] $\Gamma^{-}_{\mathcal{S}}$ [D8.2.2]				

Table A.2.2: Functions and relations between different variable types; numbers in brackets refer to the Section or Definition number

	ψ	f	$\mathfrak{c}$	$\mathfrak{m}$	$\mathfrak{p}$
f	f/ψ [D6.1.1]				
$\mathfrak{m}$	$\psi\mathfrak{m}$ [D3.2.2]				
$\mathfrak{p}$	$\mathfrak{p}\psi$ [D3.2.3]	$f(\mathfrak{p})$ [D3.2.3] $f(\mathfrak{p}, \mathfrak{t})$ [D3.2.5] $\mathrm{MNX}(f; \mathfrak{p})$ [D5.1.1] $\mathrm{NGX}(f; \mathfrak{p})$ [D5.1.3]		$\mathfrak{p} \oplus \mathfrak{m}$ [D3.2.4]	
$\mathfrak{t}$		$f(\mathfrak{p}, \mathfrak{t})$ [D3.2.5]			$t\mathfrak{p}$ [D3.2.3] $f(\mathfrak{p}, \mathfrak{t})$ [D3.2.5]
$\mathcal{Q}$		$\langle\langle \mathcal{Q}, f \rangle\rangle$ [D6.2.1]			
Γ	Γ/ψ [D6.1.1] $\psi_0 \geqslant_\Gamma \psi_1$ [D8.1.1] Γ^+_ψ [D8.2.1] Γ^-_ψ [D8.2.1]		$\mathrm{MNX}(\Gamma; \mathfrak{c})$ [D5.1.1] $\mathrm{NGX}(\Gamma; \mathfrak{c})$ [D5.1.3]		$\mathrm{MNX}(\Gamma; \mathfrak{p})$ [D5.1.1] $\mathrm{NGX}(\Gamma; \mathfrak{p})$ [D5.1.3] $\mathfrak{M}^+_\Gamma(\mathfrak{p})$ [D5.1.4] $\mathfrak{M}^-_\Gamma(\mathfrak{p})$ [D5.1.4] $\mathfrak{M}^o_\Gamma(\mathfrak{p})$ [D5.1.4]

Table A.2.3: Functions and relations between variable types; numbers in brackets refer to the Section or Definition number

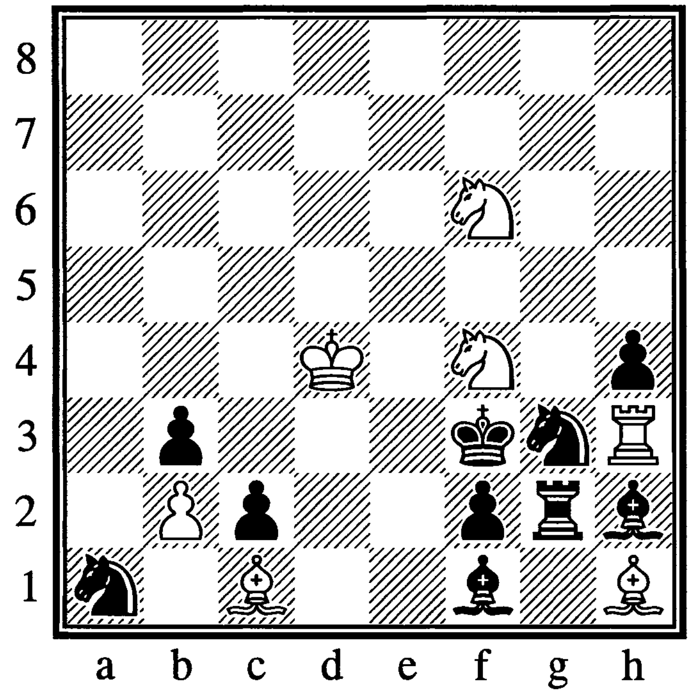


Figure A.1: Sam Lloyd's Comet.

king with the white-squared bishop. If the white king stays on black squares, then a parity argument shows that the king will always arrive at g5 at the wrong time, no matter what path was taken. The solution is that the white king needs to lose one tempo on the only white square where this can be done safely out of reach of black's white-squared bishop, namely a8.

Index

Bibliography

[1] Bruce Abramson. Learning Expected-Outcome E-valuators in Chess. In Hans Berliner, editor, *Proceedings of the AAAI Spring Symposium on Computer Game Playing*, pages 26–28. Stanford University, 1988.

[2] Bruce Abramson. Expected Outcome: A General Model of Static Evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(2):182–193, February 1990.

[3] L. Victor Allis. *Searching for Solutions in Games and Artificial Intelligence.* Ph.d. thesis, University of Limburg, Maastricht, 1994.

[4] L. Victor Allis, Maarten van der Meulen, and H. Jaap van den Herik. Proof-Number Search. *Artificial Intelligence*, 66(1):91–124, 1994.

[5] Vadim Anshelevich. The Game of Hex: An Automatic Theorem-Proving Approach to Game Programming. In *AAAI*, pages 189–194, 2000.

[6] Vadim Anshelevich. Hexy Wins Hex Tournament. *Journal of the International Computer Games Association*, 23(3):181–184, 2000.

[7] Vadim Anshelevich. A Hierarchical Approach to Computer Hex. *Artificial Intelligence*, 134(1-2):101–120, January 2002.

[8] Argimiro A. Arratia-Quesada. On the Descriptive Complexity of a Simplified Game of Hex. *Logic Journal of the IGPL*, 10(2):105–122, March 2002.

[9] Donald F. Beal and Martin C. Smith. Random Evaluations in Chess. *International Computer Chess Association Journal*, 17(1):3–9, 1994.

[10] Anatole Beck, Michael N. Bleicher, and Donald W. Crowe. *Excursions into Mathematics*, chapter Games, pages 317–387. Worth, New York, 1969.

[11] Claude Berge. L'Art Subtil du Hex. Manuscript, 1977.

[12] Claude Berge. Some Remarks about a Hex Problem. In David A. Klarner, editor, *The Mathematical Gardner*, pages 25–27. Wadsworth International, Belmont, 1981.

[13] Elwyn R. Berlekamp, John H. Conway, and Richard K. Guy. *Winning Ways for your Mathematical Plays.* Academic Press, New York, 1982. Second Edition: A.K. Peters, 2001–2004.

[14] David Berman. Hex Must Have a Winner: an Inductive Proof. *Mathematics Magazine*, 49(2):85–86, March 1976. See two letters on simpler proofs in 49(3):156, May 1976.

[15] Darse Billings. Thoughts on Roshambo. *International Computer Games Association Journal*, 23(1), March 2000.

[16] Darse Billings, Aaron Davidson, Jonathan Schaeffer, and Duane Szafron. The Challenge of Poker. *Artificial Intelligence*, 134:201–240, 2002.

[17] Bruno Bouzy and Bernard Helmstetter. Developments on Monte Carlo Go. In H. Jaap van den Herik, Hiroyuki Iida, and Ernst A Heinz, editors, *Advances in Computer Games ACG-10*, pages 159–174, Boston, 2003. Kluwer Academic Publishers.

[18] Cameron Browne. *Hex Strategy: Making the Right Connections.* A. K. Peters, Natick, Massachusetts, 2000.

[19] Bernd Brügmann. Monte Carlo Go. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:182–193, 1993.

[20] Michael Buro. Toward Opening Book Learning. *International Computer Chess Association Journal*, 22(2):98–102, 1999.

[21] David Bush, Karl Heuer, and Scott Huddleston. Personal communication, 1999.

[22] Tristan Cazenave. Abstract Proof Search. In Tony Marsland and Ian Frank, editors, *Computers and Games*, pages 39–54, Hamamatsu, 2000.

[23] John H. Conway. *On Numbers and Games.* Academic Press, New York, 1976. Second edition: A.K. Peters, 2001.

[24] Rémi Coulom. Efficient Selectivity and Backup Operators in Monte-Carlo Tree Search. In *Computers and Games*, Torino, Italy, 2006.

[25] Joseph C. Culberson. Sokoban is PSPACE-complete. Technical report, University of Alberta, 1997.

[26] Chrilly Donninger. Null Move and Deep Search: Selective Search Heuristics for Obtuse Chess Programs. *International Computer Chess Association Journal*, 16(3):137–143, 1993.

[27] Uwe Egly, Hans Tompits, and Stefan Woltran. On Quantifier Shifting for Quantified Boolean Formulas. In *Proceedings of the SAT-02 Workshop on Theory and Applications of Quantified Boolean Formulas (QBF-02)*, pages 48–61, 2002.

[28] Herbert Enderton. Infrequently asked questions about the game of Hex. Web page `http://www.cs.cmu.edu/People/hde/hex/hexfaq`.

[29] Shimon Even and Robert Endre Tarjan. A Combinatorial Problem which is Complete in Polynomial Space. *Journal of the Association for Computing Machinery*, 23:710–719, 1976.

[30] Michael R. Fellows. The Robertson-Seymour Theorems: a Survey of Applictions. *Contemporary Mathematics*, 89:1–18, 1989.

[31] Aviezri S. Fraenkel and David Lichtenstein. Computing a perfect strategy for $n \times n$ chess requires time exponential in n. *Journal of Combinatorial Theory, series A*, 31:199–214, 1981.

[32] David Gale. The Game of Hex and the Brouwer Fixed Point Theorem. *American Mathematical Monthly*, 86(10):818–827, 1979.

[33] Martin Gardner. Mathematical Games. *Scientific American*, pages 124–129, October 1958.

[34] Martin Gardner. *The Scientific American Book of Mathematical Puzzles and Diversions*, chapter The game of Hex, pages 73–83. Simon and Schuster, New York, 1959. Original column in Scientific American, July 1957, pp. 145–150. Also see August 1957 pp. 120–127, October 1957 pp. 130–138.

[35] Martin Gardner. Mathematical Games. *Scientific American*, pages 148–161, July 1961.

[36] Martin Gardner. *The Second Scientific American Book of Mathematical Puzzles and Diversions*, chapter Recreational Topology, pages 78–88. Simon and Schuster, New York, 1961.

[37] M. Garey and D. Johnson. *Computers and Intractability, a Guide to the Theory of* NP-*Completeness.* W. H. Freeman, San Francisco, 1979.

[38] Ian P. Gent, Enrico Giunchiglia, Massimo Narizzano, Andrew Rowley, and Armando Tacchella. Watched Data Structures for QBF Solvers. In Armando Tacchella and Enrico Giunchiglia, editors, *Sixth International Conference on Theory and Applications of Satisfiability Testing (SAT2003)*, pages 348–355, 2003.

[39] Ian P. Gent, Holger H. Hoos, Andrew G. D. Rowley, and Kevin Smyth. Using Stochastic Local Search to Solve Quantified Boolean Formulae. In *Ninth International Conference on Principles and Practice of Constraint Programming (CP-03)*, pages 348–362. Springer Verlag, 2003.

[40] Enrico Giunchiglia, Massimo Narizzano, and Armando Tacchella. Backjumping for Quantified Boolean Logic Satisfiability. In *International Joint Conference on Artificial Intelligence*, pages 275–281, 2001.

[41] Enrico Giunchiglia, Massimo Narizzano, and Armando Tacchella. Learning for Quantified Boolean Logic Satisfiability. In *AAAI*, pages 649–654, 2001.

[42] Richard D. Greenblatt, Donald E. Eastlake, and Stephen D. Crocker. The Greenblatt Chess Program. *Fall Joint Computing Conference Proceedings*, 31:801–810, 1967. Also in D. Levy, editor, *Computer Chess Compendium*, pages 56–66. Springer-Verlag, 1988.

[43] Andrei Grigoriev. Artificial Intelligence or Stochastic Relaxation: Simulated Annealing Challenge. In David Levy and Donald F. Beal, editors, *Heuristic Programming in Artificial Intelligence 2: the Second Computer Olympiad*, pages 210–216, Chicester, 1991. Ellis Horwood.

[44] Ryan Hayward. A Note on Domination in Hex. Technical report, University of Alberta, 2003.

[45] Ryan Hayward, Broderick Arneson, and Philip Henderson. Verifying Hex Strategies. In *Computers and Games*, Torino, Italy, 2006.

[46] Ryan Hayward, Yngvi Björnsson, Michael Johanson, Morgan Kan, Nathan Po, and Jack van Rijswijck. Solving 7×7 Hex: Virtual Connections and Game State Reduction. In H. Jaap van den Herik, Hiroyuki Iida, and Ernst A Heinz, editors, *Advances in Computer Games ACG-10*, pages 261–278. Kluwer Academic Publishers, Boston, 2003.

[47] Ryan Hayward, Yngvi Björnsson, Michael Johanson, Morgan Kan, Nathan Po, and Jack van Rijswijck. Solving 7×7 Hex with Domination, Fill-In, and Virtual Connections. *Theoretical Computer Science*, 349:123–139, 2005.

[48] Ryan Hayward and Jack van Rijswijck. Hex and Combinatorics. *Discrete Mathematics*, to appear, 2006.

[49] Ryan Hayward, Jack van Rijswijck, Yngvi Björnsson, and Michael Johanson. Dead Cell Analysis in Hex and the Shannon Game. In *Graph Theory 2004: In Memory of Claude Berge*. Birkhauser, 2005.

[50] Piet Hein. Unpublished manuscript, 1942. See `http://maarup.net/thomas/hex/`.

[51] Piet Hein. Vil de lære Polygon? *Politiken newspaper*, 26 December 1942. Subsequent articles on December 27, 28, 29, and 30.

[52] H. Jaap van den Herik, Jos Uiterwijk, and Jack van Rijswijck. Games Solved: Now and in the Future. *Artificial Intelligence*, 134(1-2):277–312, January 2002.

[53] Peter Jozef Jansen. *Using Knowledge about the Opponent in Game-Tree Search.* Ph.d. thesis, Carnegie-Mellon University, 1992.

[54] Stefan Kiefer. Die Menge der Virtuellen Verbindungen im Spiel Hex ist PSPACE-Vollständig. Technical report, Stuttgart University, 2003.

[55] Stefan Kiefer. The Set of Virtual Connections in the Game of Hex is PSPACE-Complete. Technical report, Stuttgart University, 2003.

[56] Ton Kloks and Dieter Kratsch. Listing all Minimal Separators of a Graph. *SIAM Journal on Computing*, 27(3):605–613, 1998.

[57] Donald E. Knuth and Ronald W. Moore. An Analysis of Alpha-Beta Pruning. *Artificial Intelligence*, 6(4):293–326, 1975.

[58] Dieter Kratsch and Jerry Spinrad. Between $o(nm)$ and $o(n^\alpha)$. In *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 158–167, Baltimore, 2003.

[59] Jeffrey Lagarias and Danny Sleator. Who Wins Misère Hex? In Elwyn Berlekamp and Tom Rogers, editors, *The Mathemagician and Pied Puzzler*, pages 237–240. A. K. Peters, 1999.

[60] Daniel Le Berre, Massimo Narizzano, Laurent Simon, and Armando Tacchella. The Second QBF Solvers Comparative Evaluation. In Holger H. Hoos and David G. Mitchell, editors, *Seventh International Conference on Theory and Applications of Satisfiability Testing (SAT2004)*, pages 376–392, 2005.

[61] Daniel Le Berre, Laurent Simon, and Armando Tacchella. Challenges in the QBF Arena: The SAT'03 Evaluation of QBF solvers. In Armando Tacchella and Enrico Giunchiglia, editors, *Sixth International Conference on Theory and Applications of Satisfiability Testing (SAT2003)*, pages 468–485, June 2003.

[62] Alfred Lehman. A Solution of the Shannon Switching Game. *Journal of the Society of Industrial and Applied Mathematics*, 12:687–725, 1964.

[63] H. G. Leimer. Optimal Decomposition by Clique Separators. *Discrete Mathematics*, 113:90–123, 1993.

[64] Thomas Maarup. Everything you always wanted to know about hex, but were afraid to ask. Master's thesis, University of Southern Denmark, 2005.

[65] Tony A. Marsland and Murray Campbell. Parallel Search of Strongly Ordered Game Trees. *Computing Surveys*, 14(4):533–551, 1982.

[66] David A. McAllester. Conspiracy Numbers for Minmax Searching. *Artificial Intelligence*, 35:287–310, 1988.

[67] Gabor Melis. Personal communication, 2005.

[68] Gabor Melis and Ryan Hayward. Six Wins Hex Tournament. *Journal of the International Computer Games Association*, 26(4):278–280, 2003.

[69] Gabor Melis and Ryan Hayward. Hex Gold at Graz: Six Defeats Mongoose. *Journal of the International Computer Games Association*, to appear, 2005.

[70] Massimo Narizzano, Luca Pulina, and Armando Tacchella. Report of the Third QBF Solvers Evaluation. *Journal of Satisfiability, Boolean Modeling and Computation*, 2:145–164, 2006.

[71] John Nash. Some Games and Machines for Playing Them. Technical report, RAND, 1952.

[72] John Nash. Personal communication, 1999.

[73] John von Neumann and Oskar Morgenstern. *Theory of Games and Economic Behaviour*. Princeton, 1944.

[74] Kohei Noshita. Union-Connectinos and a Simple Readable Winning Way in 7×7 Hex. In *Ninth Game Programming Workshop*, pages 72–79, Kanagawa, Japan, 2004.

[75] Axel Ostmann. Decisions by Players of Comparable Strength. *Zeitschrift für Nationalökonomie*, 45:145–159, 1985.

[76] Axel Ostmann. On the Minimal Representation of Homogeneous Games. *International Journal of Game Theory*, 16:69–81, 1987.

[77] Axel Ostmann. Order and Symmetry of Simple Games. *Note di Matematica*, 13:251–267, 1993.

[78] Yuval Peres, Scott Sheffield, Oded Schramm, and David Wilson. Random-Turn Hex and Other Selection Games. arXiv:math.PR/0508580, 2005.

[79] Rune Rasmussen and Frederic Maire. An Extension of the H-Search Algorithm for Artificial Hex Players. In *Proceedings of the 17th Australian Joint Conference on Artificial Intelligence*, Cairns, Australia, 2004.

[80] Stefan Reisch. Hex ist PSPACE-vollständig. *Acta Informatica*, 15:167–191, 1981.

[81] Jack van Rijswijck. Queenbee's home page. `http://www.cs.ualberta.ca/~queenbee`.

[82] Jack van Rijswijck. Are Bees better than Fruitflies? Experiments with a Hex Playing Program. In Howard Hamilton, editor, *Advances in Artificial Intelligence*, pages 13–25, Montreal, 1999. Canadian Society for Computational Studies of Intelligence.

[83] Jack van Rijswijck. Computer Hex: Are Bees better than Fruitflies? Master's thesis, University of Alberta, Edmonton, Canada, 2000.

[84] Jack van Rijswijck. Search and Evaluation in Hex. Technical report, University of Alberta, 2002.

[85] Jack van Rijswijck. Binary Combinatorial Games. In Richard Nowakowski, editor, *Games of No Chance 3.* to appear, 2006.

[86] John Michael Robson. n By n Checkers is Exptime Complete. *Society for Industrial and Applied Mathematics Journal on Computing*, 13(2):252–267, May 1984.

[87] The International Conferences on Theory and Applications of Satistiability Testing. `http://www.satisfiability.org`.

[88] Thomas Schaefer. On the Complexity of Some Two-Person Perfect-Information Games. *Journal of Computing System Science*, 16:185–225, 1978.

[89] Jonathan Schaeffer. The History Heuristic. *International Computer Chess Association Journal*, 6(3):16–19, 1983.

[90] Jonathan Schaeffer. The History Heuristic and Alpha-Beta Search Enhancements in Practice. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1989.

[91] Craige Schensted and Charles Titus. *Mudcrack-Y and Poly-Y.* Neo Press, Maine, 1975.

[92] Claude E. Shannon. Computers and Automata. In *Proceedings of the Institute of Radio Engineers*, volume 41, pages 1234–1241, 1953.

[93] Brian Sheppard. World Championship Caliber Scrabble. *Artificial Intelligence*, 134(1-2):241–275, January 2002.

[94] Brian Sheppard. World Championship Caliber Scrabble. In Jonathan Schaeffer and Jaap van den Herik, editors, *Chips Challenging Champions*, pages 283–317. Elsevier, 2002.

[95] Brian Sheppard. Efficient Control for Selective Simulations. *International Computer Games Association Journal*, 27(2):67–80, 2004.

[96] David J. Slate and Larry R. Atkin. Chess 4.5 - The Northwestern University Chess Program. In Peter W. Frey, editor, *Chess Skill in Man and Machine*, pages 82–118. Springer-Verlag, New York, 1977.

[97] Larry J. Stockmeyer and Ashok K. Chandra. Provably Difficult Combinatorial Games. *Journal of the Society for Industrial and Applied Mathematics*, 8(2):151–174, May 1979.

[98] Robert Endre Tarjan. Decomposition by Clique Separators. *Discrete Mathematics*, 55:221–232, 1985.

[99] Gerald Tesauro and Gregory R. Galperin. Online Policy Improvement using Monte Carlo Search. In *Advances in Neural Information Processing Systems*, pages 1068–1074. MIT Press, Cambridge, MA, 1996.

[100] Thomas Thomsen. Lambda-Search in Game Trees – with Application to Go. In Tony Marsland and Ian Frank, editors, *Computers and Games*, pages 19–38, Hamamatsu, 2000.

[101] Sue Whitesides. A Method for Solving Certain Graph Recognition and Optimization Problems with Applications to Perfect Graphs. In Claude Berge and Vašek Chvátal, editors, *Topics on Perfect Graphs*, volume 21 of *Annals of Discrete Mathematics*. North Holland, Amsterdam, 1984.

[102] Yōhei Yamasaki. Theory of Division Games. In *Publications of the Research Institute for Mathematical Sciences*, volume 14, pages 337–358. Kyoto University, 1978.

[103] Jing Yang. A winning 9x9 Hex Strategy. To appear, `http://www.ee.umanitoba.ca/~jingyang`.

[104] Jing Yang, Simon Liao, and Mirek Pawlak. A Decomposition Method for Finding Solution in Game Hex 7x7. In *International Conference on Application and Development of Computer Games in the 21st Century*, pages 96–111, Hong Kong, November 2001.

[105] Jing Yang, Simon Liao, and Mirek Pawlak. Another Solution for Hex 7x7. Technical report, University of Manitoba, Winnipeg, Canada, 2002. `http://www.ee.umanitoba.ca/~jingyang/TR.pdf`.

[106] Jing Yang, Simon Liao, and Mirek Pawlak. New Winning and Losing Positions for 7x7 Hex. In *Computers and Games*, Edmonton, 2002.

[107] Ling Zhao and Martin Müller. Game-SAT: a Preliminary Report. In *Seventh International Conference on Theory and Applications of Satisfiability Testing (SAT 2004)*, pages 357–362, Vancouver, Canada, 2004.