THÈSE

PRÉSENTÉE À

L'UNIVERSITÉ DE BORDEAUX

ÉCOLE DOCTORALE DE MATHÉMATIQUES ET D'INFORMATIQUE

par **Edon Kelmendi**

POUR OBTENIR LE GRADE DE

DOCTEUR

SPÉCIALITÉ : INFORMATIQUE

Two-Player Stochastic Games with Perfect and Zero Information

---

Date de soutenance : 02 décembre 2016

Devant la commission d'examen composée de :

Véronique Bruyère ...	Professeur, Université de Mons	Examinatrice
Laurent Doyen	Chargé de recherche, CNRS LSV-ENS Cachan	Rapporteur
François Dufour	Professeur, INP Bordeaux	Directeur de thèse
Hugo Gimbert	Chargé de recherche, CNRS LaBRI Bordeaux	Directeur de thèse
Mahesh Viswanathan	Professuer, University of Illinois	Rapporteur
Marc Zeitoun	Professeur, Université de Bordeaux	Président

– 2016 –

TITLE Two-Player Stochastic Games with Perfect and Zero Information

ABSTRACT We consider stochastic games that are played on finite graphs. The subject of the first part are two-player stochastic games with perfect information. In such games the two players take turns choosing actions from a finite set, for an infinite duration, resulting in an infinite play. The objective of the game is given by a Borel-measurable and bounded payoff function that maps infinite plays to real numbers. The first player wants to maximize the expected payoff, and the second player has the opposite objective, that of minimizing the expected payoff. We prove that if the payoff function is both shift-invariant and submixing then the game is half-positional. This means that the first player has an optimal strategy that is at the same time pure and memoryless. Both players have perfect information, so the actions are chosen based on the whole history.

In the second part we study finite-duration games where the protagonist player has zero information. That is, he gets no feedback from the game and consequently his strategy is a finite word over the set of actions. Probabilistic finite automata can be seen as an example of such a game that has only a single player. First we compare two classes of probabilistic automata: leaktight automata and simple automata, for which the value 1 problem is known to be decidable. We prove that simple automata are a strict subset of leaktight automata. Then we consider half-blind games, which are two player games where the maximizer has zero information and the minimizer is perfectly informed. We define the class of leaktight half-blind games and prove that it has a decidable maxmin reachability problem.

KEYWORDS stochastic games, half-positional, shift-invariant, submixing, probabilistic automata, leaktight automata, simple automata, half-blind games, maxmin reachability

TITRE Jeux Stochastiques à Deux Joueurs à Information Parfaite et Zéro

ABSTRAIT On considère des jeux stochastiques joués sur un graphe fini. La première partie s'intéresse aux jeux stochastiques à deux joueurs et information parfaite. Dans de tels jeux, les joueurs choisissent des actions dans ensemble fini, tour à tour, pour une durée infinie, produisant une histoire infinie. Le but du jeu est donné par une fonction d'utilité qui associe un réel à chaque histoire, la fonction est bornée et Borel-mesurable. Le premier joueur veut maximiser l'utilité espérée, et le deuxième joueur veut la minimiser. On démontre que si la fonction d'utilité est à la fois shift-invariant et submixing alors le jeu est semi-positionnel. C'est-à-dire le premier joueur a une stratégie optimale qui est déterministe et sans mémoire. Les deux joueurs ont information parfaite: ils choisissent leurs actions en ayant une connaissance parfaite de toute l'histoire.

Dans la deuxième partie, on étudie des jeux de durée fini où le joueur protagoniste a zéro information. C'est-à-dire qu'il ne reçoit aucune information sur le déroulement du jeu, par conséquent sa stratégie est un mot fini sur l'ensemble des actions. Un automates probabiliste peut être considéré comme un tel jeu qui a un seul joueur. Tout d'abord, on compare deux classes d'automates probabilistes pour lesquelles le problème de valeur 1 est décidable: les automates leaktight et les automates simples. On prouve que la classe des automates simples est un sous-ensemble strict de la classe des automates leaktight. Puis, on considère des jeux semi-aveugles, qui sont des jeux à deux joueurs où le maximiseur a zéro information, et le minimiseur est parfaitement informé. On définit la classe des jeux semi-aveugles leaktight et on montre que le problème d'accessibilité maxmin est décidable sur cette classe.

MOTS-CLÉS jeux stochastiques, demi-positionnel, shift-invariant, submixing, automates probabilistes, automates leaktight, automates simples, jeux demi-aveugles, accessibilité maxmin

LABORATOIRE D'ACCUEIL

Laboratoire Bordelais de Recherche en Informatique (LaBRI)

Unité Mixte de Recherche CNRS (UMR 5800)
351, cours de la Libération F-33405 Talence cedex,
France

RÉSUMÉ Les jeux sont des modèles mathématiques permettant de modéliser des prises de décision rationnelles. Les décideurs peuvent être en compétition ou coopérer. Le domaine d'étude qui concerne les jeux, la théorie des jeux, est déjà très bien établie. Ses applications concernent de nombreux domaines, y compris l'économie, la science politique, la psychologie, la biologie, l'informatique et la logique.

Très généralement, dans un jeu, les joueurs choisissent des actions, et en le faisant changent l'état du jeu. En fin de compte, en fonction des actions qui ont été prises et de la suite d'états par laquelle le jeu passe, chaque joueur reçoit un un certain paiement. Les joueurs visent à maximiser ce paiement.

Les questions essentielles qui sont étudiées sont les suivantes: combien les joueurs peuvent-ils gagner, quelle est la meilleure suite d'actions à prendre (stratégies), y at-il des points d'équilibre (choix de stratégie où aucun joueur n'a d'incitation à changer sa stratégie).

Au cours des dernières décennies, avec le développement de l'informatique, les problèmes algorithmiques liés à par ces questions ont naturellement gagné en importance: est-il possible de calculer combien les joueurs peuvent gagner, la meilleure stratégie, les points d'équilibre etc. et si oui, à quel point est-il difficile d'effectuer ces calculs? Cette thèse aborde ces questions.

Parce que la notion de jeu est fondamentale et ajustable à la modélisation de nombreuses situations, il existe de nombreux types de jeux qui ont des propriétés très différentes et qui nécessitent des méthodes différentes pour être étudiés.

Nous considérons des jeux qui sont: un ou deux joueurs, somme nulle, à durée finie ou infinie, discrets, stochastiques et tour à tour, qui ont un nombre fini d'états et un nombre fini d'actions pour les deux joueurs. Nous représentons le jeu par un graphe fini, dont les sommets représentent les états du jeu et dont les arcs sont les actions que les joueurs peuvent prendre.

Les jeux sont classés en fonction de certaines de leurs propriétés de base. Tout d'abord, le nombre de joueurs: un jeu peut avoir seulement un joueur, deux joueurs ou plus, un ensemble fini mais fixe de joueurs (ces jeux sont surtout utilisés en biologie). Nous regarderont les jeux à un et deux joueurs ici.

Il ya des jeux où les joueurs peuvent coopérer entre eux, et ceux où il n'ya aucune raison de le faire. Les jeux à somme nulle sont des jeux où la somme des gains que chaque joueur reçoit est toujours égale à zéro. Par exemple, dans un jeu à somme nulle, un joueur reçoit le montant exact que son adversaire perd: il reçoit x alors que son adversaire reçoit $-x$. En particulier, les jeux à deux joueurs où un joueur gagne ou perd, *i.e.* il n'y a pas de match nul, sont zéro-somme, par exemple, le papier-ciseaux-pierre est à somme nulle. Tous les jeux de cette thèse sont à somme nulles, il n'y a donc pas de coopération entre les joueurs.

Dans cette thèse, nous nous concentrons uniquement sur les jeux avec un ensemble fini d'états et un ensemble fini d'actions. Dans la littérature, les jeux avec un ensemble infini d'états et d'actions sont également étudiés, en particulier dans la théorie des ensembles descriptifs.

Les jeux que nous étudierons sont discrets, c'est-à-dire que les joueurs choisissent leurs actions à intervalle discret. Outre les jeux discrets, il ya des jeux continus aussi. Là, l'état du jeu change en continu, par exemple selon une équation différentielle.

Une autre façon de classer les jeux est de savoir si, après un joueur joue une action, l'état du jeu change d'une manière déterministe ou aléatoire, en suivant une certaine loi de probabilité. Les premiers sont des jeux déterministes, les seconds des jeux stochastiques et ils sont le sujet de cette thèse.

Dans la première partie de la thèse nous parlons de jeux à durée infinie, et dans la deuxième partie de la thèse de jeux à durée finie mais arbitraire.

Les jeux peuvent être simultanés ou joués à tour de rôle. Un jeu est simultané si les joueurs choisissent une action indépendamment de chaque autre en même temps, puis l'état du jeu est modifié en fonction de l'état actuel et de cette paire des actions. Un jeu est joué à tour de rôle si les joueurs prennent leurs décisions l'un après l'autre. Par exemple, les échecs se jouent à tour de rôle, le premier joueur fait un mouvement, puis l'autre joueur et ainsi de suite. D'autre part, dans ciseaux, papier, pierre, les deux joueurs jouent en même temps, il s'agit d'un jeu simultané. Nous ne considérerons que les jeux tour à tour.

Une autre propriété importante est l'information que les joueurs ont quand ils prennent des décisions. Dans le jeu d'échecs, par exemple, les deux joueurs ont toutes les informations qui existent, *i.e.* ils voient l'échiquier, et ils connaissent tous les deux les mouvements qui ont été faits. Nous disons qu'ils sont parfaitement informés ou qu'il s'agit d'un jeu à information parfaite. Tous les jeux ne sont pas comme ça, un contre-exemple notoire est le poker.Au poker, les joueurs n'ont qu'une vue partielle du jeu. Ils connaissent les cartes qu'ils tiennent mais pas celles de leurs adversaires. Les joueurs ne savent pas exactement l'état du jeu, mais seulement un ensemble d'états dans lequel il pourrait être. Ces jeux sont appelés jeux avec em information partiel.

En informatique et en logique, les jeux sont souvent utilisés comme un outil théorique, par exemple les jeux Ehrenfeucht-Fraïssé dans la théorie des modèles finis, les jeux de parité dans vérification, etc., mais il existe une autre utilité importante des jeux dans informatique et vérification, c'est-à-dire la synthèse du contrôleur.

On part d'un système que l'on veut contrôler. En effectuant des actions, le contrôleur change l'état du système, mais *l'environnement* peut également modifier l'état du système. Supposons que nous voulons contrôler le système tel que il n'entre jamais dans un ensemble interdit d'états, ou qu'il passe par quelque état infiniment souvent. De nombreux scénarios réels peut être vus d'une telle

manière, comme un contrôleur et environnement qui change l'état du système, tour à tour. Ceci peut être modélisé comme un jeu avec deux joueurs, contrôleur et environnement. Le contrôleur souhaite réaliser l'objectif de ne jamais entrer dans l'état interdit, ou voir un état distingué infiniment souvent, alors que l'environnement est son adversaire et essaie de faire le contraire, *e.g.* à entrer dans l'état interdit. Le problème de savoir si le contrôleur peut atteindre son objectif est la question de savoir si Controller a une stratégie gagnante dans un tel jeu. En construisant efficacement la stratégie, nous synthétisons ce que le contrôleur du système devrait faire pour atteindre son objectif, indépendamment de l'environnement.

Cette application centrale soulève l'importance des jeux à information partielle. En effet, dans de nombreux cas, le contrôleur ne peut pas connaître l'état exact du système, par exemple il est aidé par des capteurs qui sont défectueux parfois, ou le système est contrôlé par une interface qui ne révèle pas tous ses rouages internes (ce qui est en fait la norme).

Cependant, les jeux à informations partielle sont beaucoup plus difficiles à calculer que leurs équivalents à information parfaits. Dans plusieurs cas, un problème qui est relativement facile de décider pour les jeux à information parfaite, devient indécidable sur les jeux à informations partielle.

Comme il est habituellement fait en mathématiques, afin de faire la lumière sur le cas général, des cas spéciaux sont considérés, nous allons faire de même dans cette thèse. Dans le but de comprendre un peu mieux les jeux à information partielle, nous étudier deux cas extrêmes spéciaux, l'information parfaite, où les joueurs connaissent l'état exact du jeu, et l'information zéro, où le joueur protagoniste n'a aucune information sur l'état du jeu. En d'autres termes, il joue les actions aveuglément, sans obtenir aucune indication sur l'état du jeu.

La thèse est divisée en deux parties, la première partie s'intéresse aux jeux à deux joueurs et information parfaite tandis que la deuxième partie s'intéresse aux jeux information zéro.

Jeux à information parfaite.

Dans les jeux stochastiques à deux joueurs et information parfaite, l'ensemble des états est divisé en deux, l'ensemble des états qui sont contrôlés par le joueur 1, et l'ensemble des états qui sont contrôlés par son adversaire, le joueur 2.

Si le jeu est dans un état qui est contrôlé par le joueur 1 (joueur 2), alors c'est le joueur 1 (joueur 2) qui choisit une action. Suite à cette action, l'état du jeu change en fonction d'une distribution de probabilités sur l'ensemble des états, qui dépend de l'état actuel et de l'action qui a été prise par le joueur 1 (joueur 2). Ainsi, le jeu se déroule, produisant une séquence infinie de paires action, état, appelée l'histoire.

La fonction de paiement, associe à cette séquence de paires états action un nombre réel x. Le joueur 1 reçoit le montant x et son adversaire reçoit le montant $-x$. Le joueur 1 préfère les histoires qui donnent des paiements plus grands,

alors que son adversaire les histoires qui donnent des paiements plus petits, les joueurs sont appelés le maximiseur et le minimiseur, respectivement.

A ce degré de généralité, les jeux comme celui que nous avons décrit ci-dessus ne possèdent pas de propriétés algorithmiques intéressantes, celles-ci commencent à émergent uniquement sous certaines restrictions de la fonction de paiement. Notamment, si la fonction de paiement est Borel-mesurable et bornée, alors ces jeux stochastiques ont des valeurs [Martin, 1998]. Cela signifie qu'il existe un point d'équilibre. Plus précisément, cela signifie que le *minmax* de paiement attendu est égal au *maxmin* de paiement attendu, cette quantité est appelée la valeur du jeu. La valeur minimale de paiement attendu est l'infimum sur les stratégies du minimiseur, du supremum sur les stratégies du maximiseur, de paiement attendu dans le cadre des stratégies choisies; Symétriquement pour maxmin. Cela nous permet de parler de stratégies ϵ-optimales, *i.e.* des stratégies qui assurent que le paiement attendu est à une distance ϵ de la valeur du jeu.

Dans les jeux à information parfaite, les stratégies qui sont à la disposition des joueurs sont des fonctions qui mappent les histoires finies du jeu (séquences finies des paires état action) à une action, ou potentiellement une distribution de probabilités sur l'ensemble des actions, selon laquelle l'action est choisie.

Beaucoup de jeux comme ci-dessus ont été étudiés dans la littérature, y compris les jeux de moyenne-payoff, les jeux à prix réduit, les jeux de parité, certains jeux de compteur etc. Le théorème de Martin garantit que tous ces jeux ont des stratégies ϵ-optimales pour tous $\epsilon > 0$, mais pas nécessairement de stratégies optimales (stratégies 0-optimales). Mais comme il s'avère, dans de nombreux jeux classiques, y compris ceux que nous avons mentionnés ci-dessus, les deux joueurs possèdent des stratégies optimales. Plus intéressant, les stratégies sont très simples. En fait, ils ont des stratégies optimales les plus simples possibles: des stratégies positionnelles.

En général, les stratégies sont des objets compliqués, la quantité d'information dont les joueurs ont besoin est potentiellement infinie, mais pour beaucoup de jeux intéressants, les stratégies optimales n'ont pas besoin d'avoir de mémoire. Les stratégies sont appelés positionnelles si elles n'utilisent ni la mémoire ni la puissance de la randomisation, ce sont des stratégies qui toujours choisissent la même action unique en fonction de l'état du jeu, *i.e.* ce sont des fonctions de l'ensemble des états vers l'ensemble des actions.

Les jeux dans lesquels les deux joueurs ont toujours des stratégies optimales et positionelles sont appelés positionnels, Les jeux dans lesquels le maximiseur a toujours une stratégie positionnelle optimale sont appelés semi-positionnels.

La positionnalité et la semi-positionnalité sont des propriétés importantes, d'autant plus que cela implique que seul un ensemble fini de stratégies doivent être prises en compte lors de la recherche de la stratégie optimale. Ceci avec le fait que beaucoup des jeux sont positionnels, soulève la question de savoir si

nous pouvons déterminer sous quelles conditions sur la fonction de paiement un jeu est (semi) positionel. Une réponse à cette question donnerait une preuve unifiée de la positionnalité; au lieu de comprendre pourquoi un jeu particulier est (semi) positionnel, nous comprendrions quelle est la raison commune qui garantit la (semi) positionnalité.

Dans le travail dans la littérature qui suit cette direction de la recherche, se présentent deux conditions qui sont imposées à la fonction de paiement: shift-invariance et submixing.

Une fonction de paiement est shift-invariant (invariant par décalage) si le déplacement de l'histoire infinis en supprimant n'importe quel préfixe fini ne change pas la valeur de la fonction. Une fonction de paiement est submixing si pour chaque trois histoires infinies de paires état action x, y, z, tel que z est un shuffling de x et y, le paiement de z est au plus le maximum de paiement de x et le paiement de y.

Dans [Gimbert, 2007], Gimbert prouve que dans les jeux stochastiques à un joueur (processus de décision Markov), si la fonction de paiement est à la fois submixing et shift-invariant alors le joueur dispose d'une strategie optimale positionnelle. Pour les jeux déterministes de deux joueurs, avec des fonctions de paiement qui mappent à $\{0, 1\}$, Kopczyński prouve que si le paiement est shift-invariant et submixing, alors le jeu est semi-positionnel en [Kopczyński, 2006, 2009].

Nous généralisons ces résultats dans [Gimbert and Kelmendi, 2014b] en prouvant que dans les jeux stochastiques à deux joueurs, si la fonction de paiement est à la fois shift-invarant et submixing, alors le jeu est semi-positionnel, *i.e.* le maximiseur a une stratégie optimale qui est positionnelle.

Jeux à information zéro.

La deuxième partie de cette thèse se caractérise par le joueur protagoniste, le maximiseur, ayant zéro information.

Les objets que nous étudierons, tout comme dans la première partie, sont des jeux stochastiques joués sur des graphes finis. Les différences seront les suivantes. Dans la première partie, nous traiterons des jeux de durée infinie, tandis que dans la seconde, les jeux auront une durée finie mais arbitraire.

Dans la première partie, le maximiseur, est parfaitement informé. Cela signifie qu'il peut fonder ses décisions sur toute l'histoire du jeu jusqu'à ce point. Il choisit parmi les stratégies qui sont des fonctions qui mappent des histoires finies à une action.

Dans un jeu à information partielle, l'histoire finie ne constitue pas une suite finie d'états par lesquels le jeu a passé du début, mais plutôt une suite finie d'ensembles d'états. Donc, le joueur ne connaît pas l'état réel du jeu, mais il connaît un ensemble d'états, auquel appartient l'état réel du jeu. Les jeux d'informations parfaites sont les jeux où les ensembles qui sont révélés au joueur

sont des singletons. Les jeux d'information zéro (ou jeux aveugles) sont les jeux où l'ensemble qui est révélé au joueur est l'ensemble complet des états.

Nous allons considérer deux jeux avec information zéro, d'abord le plus simple: les automates finis probabilistes, que nous considérons comme un jeu à un joueur et information zéro avec objectifs d'accessibilité. Deuxièmement, la généralisation d'automates probabilistes aux jeux à deux joueurs avec des objectifs d'accessibilité où l'adversaire est parfaitement informé, que nous appelons jeux semi-aveugles.

Les automates finis probabilistes (AFP) sont une généralisation d'automates finis déterministes (AFD) sur les mots, introduits pour la première fois par Rabin dans [Rabin, 1963]. Un AFD, après avoir lu une lettre, se transforme de façon déterministe en un nouvel état. Quand un AFP lit une lettre, l'état suivant est choisi selon une distribution de probabilité sur l'ensemble des états qui dépend sur l'état actuel et la lettre lue. Un AFD accepte un mot ou pas, alors qu'un automate probabiliste accepte un mot avec une certaine probabilité.

Les automates probabilistes ont de nombreuses applications, mais malheureusement la majorité des problèmes sont indécidable. Par exemple, le problème du vide est indécidable [Paz, 1971]. C'est le problème qui demande s'il existe un mot qui est accepté avec une probabilité au moins $1/2$.

Le problème que nous étudions est le *problème de la valeur 1*. On demande si pour tout $\epsilon > 0$ il existe un mot fini qui est accepté avec une probabilité d'au moins $1 - \epsilon$. Ce problème est indécidable [Gimbert and Oualhadj, 2010].

Néanmoins, des efforts récents ont été déployés pour identifier des classes intéressantes d'automates où le problème de la valeur 1 est décidable. Notamment la classe des automates leaktight dans [Fijalkow *et al.*, 2012], et la classe des automates simples dans [Chatterjee and Tracol, 2012], qui inclut toutes les autres classes connues d'automates avec problème de valeur 1 décidable. En outre, ces deux papiers utilisent des techniques qui sont tout à fait différentes pour atteindre leurs conclusions.

Nous allons prouver que la classe des automates simples est incluse strictement dans la classe des automates leaktight [Fijalkow *et al.*, 2015].

Les jeux semi-aveugles sont des jeux stochastiques à deux joueurs qui sont joués sur un graphe fini avec l'objectif d'accessibilité. Le maximiseur essaie d'atteindre l'ensemble des états finaux mais il a zéro information (il est aveugle) comme le joueur dans un automate fini probabiliste et joue de même un mot fini. Son adversaire, le minimiseur, est parfaitement informé.

Le problème que nous considérons est celui de l'accessibilité maxmin. On demande si pour tout $\epsilon > 0$ il existe une stratégie pour le maximiseur (*i.e.* un mot fini) de sorte que contre toutes les stratégies du minimiseur la probabilité d'atteindre l'ensemble des états finaux est d'au moins $1 - \epsilon$.

Le cas particulier d'un jeu semi-aveugle où le minimiseur n'a jamais de choix, est un automate probabiliste et le problème d'accessibilité maximin se réduit au

problème de la valeur 1. Puisque ce dernier est indécidable, le premier est aussi indécidable.

C'est la raison principale pour laquelle ces problèmes d'accessibilité limite-sûrs ne sont pas considérés dans la littérature, pour les jeux informations partielles, car même pour le cas le plus simple d'automates probabilistes, il est indécidable. La situation est différente pour l'accessibilité presque-sûre (positif) [Bertrand *et al.*, 2009; Nain and Vardi, 2013; Chatterjee *et al.*, 2012]. C'est la question de savoir s'il existe une stratégie pour le maximiseur de sorte que contre toutes les stratégies du minimisuer, la probabilité d'atteindre l'ensemble des états finaux est 1 (positive).

Pour les jeux semi-aveugles nous prouvons que le problème d'accessibilité maxmin est décidable dans une classe de jeux que nous appelons des jeux semi-aveugles leaktight [Kelmendi and Gimbert, 2016].

CONTENTS

1 INTRODUCTION 3
1.1 Stochastic Games on Finite Graphs . . . 3
1.2 Games with Perfect Information . . . 6
1.3 Games with Zero Information . . . 8
1.3.1 Probabilistic automata . . . 8
1.3.2 Half-blind games . . . 9

i GAMES WITH PERFECT INFORMATION 11
2 SUBMIXING AND SHIFT-INVARIANT GAMES 13
2.1 Overview . . . 14
2.2 Two Player Stochastic Games . . . 20
2.2.1 Strategies . . . 22
2.2.2 Probability measure . . . 23
2.2.3 Values and optimal strategies . . . 24
2.2.4 A Half-positional example . . . 26
2.2.5 Martingales . . . 27
2.3 Shift-Invariant and Submixing Payoff Functions . . . 30
2.4 Existence of ϵ-subgame-perfect Equilibria . . . 31
2.4.1 Weaknesses and the reset strategy . . . 34
2.4.2 Locally-optimal strategies . . . 36
2.4.3 Finitely many resets . . . 43
2.4.4 The reset strategy is 2ϵ-subgame-perfect . . . 50
2.5 Half-Positionality . . . 55
2.6 Examples . . . 60
2.6.1 Comments . . . 64
2.7 Conclusion . . . 65

ii GAMES WITH ZERO INFORMATION 67
3 PROBABILISTIC AUTOMATA 69
3.1 Overview . . . 70
3.2 Decidable Classes . . . 77
3.2.1 Leaktight automata . . . 79
3.2.2 Simple automata . . . 86
3.3 Leaktight Automata are a Superset . . . 90
3.4 Stamina, the tool . . . 98
4 HALF-BLIND LEAKTIGHT GAMES 101
4.1 Overview . . . 102

4.2 The Belief Monoid . . . 107
4.2.1 Definitions and values . . . 107
4.2.2 An Example . . . 110
4.2.3 Deterministic strategies for Min suffice . . . 111
4.2.4 The belief monoid algorithm . . . 112
4.2.5 The extended belief monoid . . . 117
4.3 Leaks . . . 118
4.4 k-decomposition Trees . . . 121
4.5 Soundness . . . 128
4.6 Completeness . . . 137
4.7 Power of Strategies and Examples . . . 142
4.8 Conclusion . . . 150
Bibliography 153

ACKNOWLEDGMENTS

Finishing this thesis would have been impossible without the help of many people. Firstly, I would like to express my deep gratitude to my advisers François Dufour and Hugo Gimbert, for their continuous support throughout the years. Everything that I know about this subject, I have learned from Hugo. I would like to thank him primarily for his immense patience, as well as for his guidance, and support — in every sense of the word. In retrospect, choosing to do an internship with Hugo, was an excellent decision whose natural continuation was this resulting thesis.

My sincere thanks also goes to the members of the jury. Especially to Laurent Doyen and Mahesh Viswanathan, for the time that they have spent to read this thesis and for their very helpful comments and suggestions. I thank Véronique Bruyère and Marc Zeitoun for agreeing to take part in the jury of my thesis.

I am tremendously grateful to the members of the formal methods group at LaBRI. Most have been my teachers. They introduced me to this part of computer science, for which I am very thankful. I will never forget their continuous support that started with the first day I came here, as well as their patience with a foreigner that did not speak French.

I would like to also thank my colleagues, my fellow office mates, and especially my collaborators Nathanaël, Youssouf, and Denis.

When one moves to a different country, one has to form a new social circle. During these years I have met a variety of very kind people, to whom I am grateful. The moments that we shared, will forever be cherished, they are some of my best memories. On the other side, the connections that we keep despite moving abroad are just as important. On this note I would like to thank my friends Betim and Eni.

Last but not least I want to express my sincere gratitude to my family for their support in my life. My parents, my sister Arta, and her family, as well as my cousins. *Ky punim iu dedikohet juve.*

1

INTRODUCTION

1.1 STOCHASTIC GAMES ON FINITE GRAPHS

Games are mathematical models through which we study logical decision making. They model competition and cooperation between the decision-makers or players. The field of study that concerns games, game theory, is already very well-established. Its applications are numerous, in many fields, including economics, political science, psychology, biology, computer science, logic. The usefulness of games need not be argued.

Very generally, in a game, the players take actions, and by doing so change the state of the game. In the end, as a function of the sequence of states through which the game went, and the actions that were taken, each player gets paid a certain amount. The players aim to increase this amount.

The essential questions that are studied are as follows: how much can the players gain, what is the best sequence of actions (strategy) to take, are there equilibrium points (strategy choices where no player has an incentive to change his strategy), among others.

In the last few decades, with the development of the electronic computer, questions concerning the computation of the above have naturally gained importance: is it possible to compute how much can the players gain, the best strategy, or one that is close to it, the equilibrium points etc. and if yes, how difficult it is to perform those computations. This thesis addresses these questions.

Because the notion of a game is fundamental and adjustable to modeling many situations, there are numerous types of games that have very different properties and that require different methods to study.

We will consider games that are: one or two player, zero-sum, finite or infinite duration, discrete, stochastic and turn-based, that have a finite number of states and a finite number of actions for both players. We will depict the game using a finite graph, whose vertices represent the states of the game and whose edges are the actions that the players can take.

Games are classified as a function of some of their basic properties. First, the number of players: a game can have only a single player, two, or more players, a finite but fixed set of players (such games are especially used in biology). We will regard one and two player games here.

There are games where the players can cooperate between themselves, and ones where there is no reason to. Zero-sum games are games where the sum of the payoffs that every player receives is always equal to zero. For example in a two player zero-sum game, a player gets paid the exact amount that his opponent loses: he receives x while his opponent receives $-x$. In particular two player games where a player either wins or loses, *i.e.* there is no draw, are zero-sum. All the games in this thesis are zero-sum, so there is no cooperation between the players.

Here we will solely focus on games with a finite set of states and a finite set of actions. In the literature, games with infinite state and action space are studied as well, especially in descriptive set theory.

The games that we will study are all discrete, that is, at discrete time steps the players take actions and change the state of the game. Besides discrete games, there are continuous games too. There the state of the game changes continuously, *e.g.* according to some differential equation. Think of the difference in modeling a continuous game such as tennis and discrete game such as any board game.

Another way of classifying games is whether after a player plays an action the state of the game changes deterministically or randomly, following some probability distribution. The former are called deterministic games, while the latter are called stochastic games. Consider the difference between tic-tac-toe, a deterministic game, and any board game that involves throwing a die. The subject of this thesis are stochastic games.

We will talk about both games of infinite duration (in the first part of the thesis), as well as games of finite but arbitrary long duration (in the second part of the thesis).

Games can be concurrent or turn-based. A game is concurrent if the players both choose an action independently of each other at the same time, and then the state of the game is changed as a function of the current state and this pair of actions. It is turn-based if the players take turns to make their decisions. For example, chess is turn-based, first one player makes a move, then the other player and so on. On the other hand in rock-paper-scissors both players make the move at the same time, by choosing either rock paper or scissors, this is a concurrent game. We will regard only turn-based games.

Another important property is the information that the players have when they make decisions. In the game of chess, for example, both players have all the information that there is, *i.e.* they both see the board, and they both know all the moves that have been made. We say that they are *perfectly informed* or that

it is a game of perfect information. But not all games are like this, for example poker is not. In poker the players have only a partial view of the game. They know the cards that they are holding but not those of their opponents. The players do not know the exact state of the game, but only some set of states in which it might be. Such games are called games with *partial information*. Their study is lucrative for the following reason.

In computer science and logic, games are often used as a theoretical tool, for example Ehrenfeucht–Fraïssé games in finite model theory, parity games in verification etc. but there is another important utility of games in computer science and verification, that is controller synthesis.

We are given a system that we want to control. By performing actions the controller changes the state of the system, but the *environment* can change the state of the system as well. Suppose that we want to control the system such that it never enters some forbidden state, or that it passes through some state infinitely often. Many real-world settings can be seen in such a way, as a controller and the environment taking turns to change the state of the system. This can be modeled as a game with two players, Controller and Environment. Controller wants to fulfill the objective of never entering the forbidden state or seeing some distinguished state infinitely often, while the environment is his opponent and tries to do the opposite, *e.g.* to enter the forbidden state. The problem of whether the controller can achieve his objective is the question of whether Controller has a winning strategy in such a game. By effectively constructing this strategy, we synthesize what the controller of the system should do to achieve his objective, whatever it might be, regardless of the environment.

This central application raises the importance of partial-information games. This is because in many scenarios the controller cannot know the exact state of the system, for example it is helped by sensors that are faulty at times, or the system is controlled through some interface which does not reveal all its inner workings (which in fact is the norm).

However, games with partial information are computationally much harder than their perfect information counterparts, as well as less understood. In a few cases, a problem that is relatively easy to decide for perfect information games, becomes undecidable on the setting of partial information games.

As it is usually done in mathematics, in order to shed light into the general case, special cases are considered, we are going to do the same in this thesis. In the quest to understand partial information games a little bit better, we will study two extreme special cases, the perfect information one, where the players know the exact state of the game, and the zero information one, where the protagonist player has no information on the state of the game. In other words he plays the actions blindly, without getting any feedback from the game.

Consequently the thesis is divided into two parts, the first part involves two player games of perfect information while the second part has to do with games of zero information.

1.2 GAMES WITH PERFECT INFORMATION

In the two player stochastic games with perfect information, the set of states is partitioned into two, the set of states that are controlled by, say, player 1, and the set of states that are controlled by his opponent, player 2.

If the game is in a state that is controlled by player 1 (player 2), then it is player 1 (player 2) that picks an action. Following this action, the state of the game changes according to a probability distribution on the set of states, which depends on the current state and the action that was taken by player 1 (player 2). Thus the game unfolds, producing an infinite sequence of state - action pairs, called the outcome.

The payoff function, then, maps this sequence of state-action pairs to a real number x. Player 1 is paid the amount x and his opponent is paid the amount $-x$. Player 1 prefers outcomes that give a larger payoff, whereas his opponent outcomes that give smaller payoffs, hence the players are suitably called the maximizer and the minimizer respectively.

In this generality, games like the one that we described above do not posses any interesting properties, those start to emerge only under some restrictions of the payoff function. Notably, if the payoff function is Borel-measurable and bounded, then such two player stochastic games have values [Martin, 1998]. This means that there exists an equilibrium point. More precisely it means that the *minmax* of the expected payoff is equal to the *maxmin* of the expected payoff, this quantity is called the value of the game. The minmax of the expected payoff is the minimum over the strategies of the minimizer, of the maximum over the strategies of the maximizer, of the expected value of the payoff function under the chosen strategies; symmetrically for maxmin. This allows us to talk about ϵ-optimal strategies, *i.e.* strategies that ensure that the expected payoff is within some ϵ distance from the value of the game.

In perfect information games the strategies that are at the disposal of the players are general behavioral ones: functions that map finite histories of the game (finite sequences of state - action pairs) to an action, or potentially a probability distribution over the set of actions, according to which the action is chosen.

Many games like above have been studied in the literature, including mean-payoff games, discounted games, parity games, some counter games etc. Martin's theorem ensures that all these games have ϵ-optimal strategies for all $\epsilon > 0$, but not optimal strategies (0-optimal strategies). But as it turns out, in many classical games, including the ones that we mentioned above both play-

ers posses optimal strategies. More interestingly the optimal strategies are very simple. In fact they are the simplest possible strategy: a positional strategy.

In general, strategies are complicated objects, the amount of information that they need to retain is potentially infinite, but for a lot of interesting games, the optimal strategies do not need to have any memory. Strategies are called positional if they do not use neither memory nor the power of randomization, they are strategies that at every time pick the same single action depending on what state the game is, *i.e.* they are functions from the set of states to the set of actions.

Games in which both players always have positional optimal strategies are called positional, while games in which one player always has a positional optimal strategy are called half-positional.

Positionality and half-positionality are important properties, especially since it implies that only a finite set of strategies should be considered when searching for the optimal strategy. This together with the fact that many important games are positional, raises the question of whether we can ascertain under what conditions on the payoff function is the game (half) positional. Furthermore it would give a unified proof of positionality; instead of understanding the reason why each game is (half) positional by itself, we would understand what is the common reason that is making all of them positional.

In the work in the literature that follows this direction of research, turn up two conditions that are imposed on the payoff function, shift-invariance and being submixing.

A payoff function is shift-invariant if shifting the infinite sequence by deleting any finite prefix does not change the value of the function. A payoff function is submixing if for any three infinite sequences of state - action pairs x, y, z, such that z is a shuffling of x and y, the payoff of z is at most the maximum of the payoff of x and the payoff of y.

In [Gimbert, 2007], Gimbert proves that for one player stochastic games (Markov decision processes), if the payoff function is both submixing and shift-invariant then the player has an optimal positional strategy. For deterministic two player games, with payoff functions that map to $\{0, 1\}$, Kopczyński proves that if the payoff function is shift-invariant and submixing, then the game is half-positional in [Kopczyński, 2006, 2009].

We further generalize these results in [Gimbert and Kelmendi, 2014b] by proving that for two player stochastic games, if the payoff function is both shift-invariant and submixing then the game is half-positional, *i.e.* the maximizer has an optimal strategy that is positional.

1.3 GAMES WITH ZERO INFORMATION

The second part of this thesis is characterized by the protagonist player, the maximizer, having zero information.

The objects that we will study, just like in the first part, are stochastic games played on finite graphs. The differences will be as follows. In the first part we will deal with games of infinite duration, while in the second, the games will have a finite but arbitrary duration.

In the first part, the maximizer, is perfectly-informed. This means that he can base his decisions on the whole history of the game up to that point. He chooses among behavioral strategies: functions that map finite histories to a mixed action.

In a partial-information game, the finite history that the players have at hand before making the decision on what action to take, does not constitute of a finite sequence of states through which the game has passed from the beginning, but rather a finite sequence of sets of states. So the player does not know the actual state of the game, but knows a set of states, to which the real state of the game belongs. Games of perfect information are special cases where the sets that are revealed to the player are singletons. Games of zero information (or blind games) are the special cases where the set that is revealed to the player is the full set of states.

We will consider two games with zero information, first the simplest one: probabilistic finite automata, which we see as one player zero information games with reachability objectives. Second, the generalization of probabilistic finite automata to two player games with reachability objectives where the opponent is perfectly informed, that we call half-blind games.

1.3.1 *Probabilistic automata*

Probabilistic finite automata are a generalization of deterministic finite automata on words, first introduced by Rabin in [Rabin, 1963]. A DFA, after reading a letter, transitions deterministically into a new state. After a PFA reads a letter, the next state is chosen according to a probability distribution on the set of states which depends on the current state and the letter that is read. A DFA either accepts a word or it does not, while a probabilistic automaton accepts a word with certain probability.

Probabilistic automata have numerous applications, but unfortunately the majority of the problems related to them are undecidable. For example, the emptiness problem is undecidable [Paz, 1971]. This is the problem that asks whether there exists a word that is accepted with probability at least $1/2$.

The problem that we are going to study is the *value 1 problem*. This asks whether for all $\epsilon > 0$ there exists a finite word that is accepted with probability at least $1 - \epsilon$. This problem is undecidable [Gimbert and Oualhadj, 2010].

Nevertheless there have been some recent efforts to identify interesting classes of automata for which the value 1 problem is decidable. Notably the class of leaktight automata in [Fijalkow *et al.*, 2012] and simple automata in [Chatterjee and Tracol, 2012], that subsumed all other known classes of automata with decidable value 1 problem. Moreover these two papers use techniques that are quite different to reach their conclusions.

We will prove that the class of leaktight automata is a strict superset of the class of simple automata [Fijalkow *et al.*, 2015].

1.3.2 *Half-blind games*

Half-blind games are two player stochastic games that are played on a finite graph with the reachability objective. The maximizer tries to reach the set of final states but he has zero information (he is blind) as the single player in a probabilistic finite automaton, and similarly plays a finite word. His opponent, the minimizer, is perfectly informed.

The problem that we will consider is that of the maxmin reachability. This asks whether for all $\epsilon > 0$ there exists some strategy for the minimizer (*i.e.* a finite word) such that against all the strategies of the minimizer the probability to reach the set of final states is at least $1 - \epsilon$.

The special case of a half-blind game where the minimizer never has any choice, is a probabilistic automaton, and the maxmin reachability problem reduces to the value 1 problem. Since the latter is undecidable the former is as well.

This is the main reason why these limit-sure reachability problems are not considered in the literature, for stochastic games with partial information, because even for the simplest case of probabilistic automata it is undecidable. The situation is different for the almost-sure (positive) reachability[Bertrand *et al.*, 2009; Nain and Vardi, 2013; Chatterjee *et al.*, 2012]. This is the question of whether there exists a strategy for the maximizer such that against all the strategies of the minimizer, the chance of reaching the set of final states is 1 (positive).

For half-blind games we will prove that the maxmin reachability problem is decidable in a class of games that does not exhibit any leaks that we call leaktight half-blind games [Kelmendi and Gimbert, 2016].

Part I

GAMES WITH PERFECT INFORMATION

2

SUBMIXING AND SHIFT-INVARIANT GAMES

The subject of this chapter is two-player stochastic games with perfect information. As in the rest of this thesis these games are played on a finite arena. An arena is a graph whose vertices (states) are partitioned into two sets, corresponding to the states controlled by the two players. If the game is in a state controlled by Player i, then it is Player i that chooses where the game moves next. He can base this choice on the whole history (he has perfect recall). The objectives of the game are given by a payoff function; a function that maps infinite histories to real numbers. One player's objective is to maximize the expectation of the payoff function while the other player tries to minimize it: it is a zero-sum game. Under some conditions on the payoff function these games admit optimal strategies. In computer science we are not interested in merely optimal strategies, but in optimal strategies that are also simple. This is because a strategy in general is a very complicated object, it might even have unbounded memory.

Fortunately, many interesting games admit very simple optimal strategies, ones that do not use randomization and whose choices do not depend on the history, but only on the current state of the game. We say that these strategies are pure and stationary. In this chapter we consider the question: *when does a player have an optimal strategy that is both stationary and pure?* We will give a sufficient condition that is relatively easy to check, under which a game will admit optimal strategies that are both stationary and pure for the player whose objective is to maximize the expected payoff. Furthermore we will see that many classical stochastic games fulfill this condition.

2.1 OVERVIEW

We will give an informal description of the games studied in this chapter, and an overview of related literature on this subject. Followed by precise definitions in the next section.

The arena is where the stochastic game takes place, it is graphically represented as in Figure 1. There is a set of states $\mathbf{S} = \{s_0, s_1, s_2, s_3, s_4\}$, partitioned into states that are controlled by Player 1: $\mathbf{S_1} = \{s_0, s_2, s_4\}$ and those controlled by Player 2: $\mathbf{S_2} = \{s_1, s_3\}$. In the graph they are represented by circles and squares respectively. There is a set of actions $\mathbf{A} = \{a, \alpha, \beta\}$ as well, and for each state $s \in \mathbf{S}$, a subset of actions $\mathbf{A}(s) \subseteq \mathbf{A}$ available in s. For example $\mathbf{A}(s_1) = \{\alpha, \beta\}$. If for some state $s \in \mathbf{S}$ there is some action $a \in \mathbf{A}$ such that $a \notin \mathbf{A}(s)$, in the graphical representation we do not draw any outgoing edge from s labeled by a, *e.g.* in Figure 1, s_1 does not have any outgoing edge labeled by a.

Say the game starts in state s_0, it is a state controlled by Player 1 so he chooses the action a and the game moves to either state s_1 or s_3, with equal probability. Suppose it goes to state s_1, this is a state controlled by Player 2, hence he has a choice between the action α and β, or he can randomize the two, *e.g.* by playing action α with probability $\frac{1}{2}$ and β with probability $\frac{1}{2}$. In that case in the next turn the game will be in one of the states s_0, s_2, s_4 with respective probabilities $\frac{1}{2}, \frac{1}{6}, \frac{1}{3}$. In this way players take turns choosing actions available in the states that are controlled by them, and build an infinite play — a word in the language $\mathcal{H} = (\mathbf{SA})^{\omega}$[1].

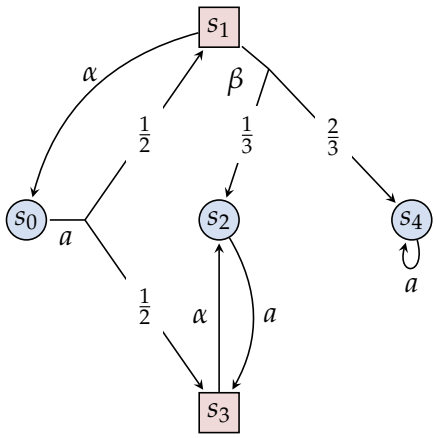


Figure 1: Arena of a stochastic game

1 this is notation from automata and formal language theory meaning the set of infinite words of the form state,action,state,action... and so on

A payoff function is a function f mapping $\mathcal{H}$ to $\mathbb{R}$. The amount that is paid to Player 1 is given by f, and the amount that is paid to Player 2 is given by $-f$. Such games are called *zero-sum*. So Player 1 prefers infinite histories with larger payoff whilst Player 2 prefers the opposite. For this reason we will call Player 1 the *maximizer* and Player 2 the *minimizer*.

When studying games, we are normally interested in optimal strategies — recipes that show how a player can play the game in an optimal manner. Under the weak assumption that the payoff function f is Borel-measurable, according Martin's second determinacy theorem [Martin, 1998] there exist ϵ-optimal strategies for all $\epsilon > 0$. These are strategies that get arbitrarily close to optimal. If one's aim is to construct such strategies, the bad news is that they might be infinite-memory strategies and hence complicated, but the good news is that for many interesting payoff functions there are optimal strategies that are very simple: they do not use memory or randomization. Such stationary and pure strategies are maps from states $\mathbf{S}$ to actions $\mathbf{A}$. If a game admits optimal strategies that are stationary and pure for both players, we say that it is *positional*, if the maximizer has an optimal strategy that is stationary and pure we say that the game is *half-positional*. In most scenarios we are interested in synthesizing the optimal strategy of one of the players (the protagonist) while the other player plays the role of the environment. Therefore knowing that the game in question is half-positional is crucially helpful. In this chapter we will give a condition of when this is true.

Usually in the literature (half) positionality is proved for a specific payoff function. Recently there has been some effort in a more general and unifying approach of giving conditions on the payoff function under which the game is (half) positional. We give a brief overview of some of the results.

In the seminal paper [Shapley, 1953], Shapley considers stochastic games with the discount payoff function. In discounted games, to every state $s \in \mathbf{S}$ we associate an immediate reward $r(s) \in \mathbb{R}$ and a discount factor $\lambda(s) \in [0, 1)$. The payoff for a history $s_0 a_0 s_1 a_1 \cdots \in \mathcal{H}$ is

$$\begin{aligned} f_{disc}(s_0 s_1 \cdots) &= r(s_0) + \lambda(s_0) r(s_1) + \lambda(s_0)\lambda(s_1) r(s_2) + \cdots \\ &= r(s_0) + \sum_{i=1}^{\infty} \Big(\prod_{j=0}^{i} \lambda(s_j)\Big) r(s_i). \end{aligned}$$

Shapley considers *concurrent* games: where players play their actions at the same time in every turn. Turn-based games, subject of this thesis, are a special case of concurrent games. Using a fixpoint of an operator approach, Shapley proves that these games have values and that they admit optimal strategies that are stationary. Although in general the optimal strategies might not be pure, because the game is concurrent. Shapley goes on to observe in the Examples and Applications section, point 3, that turn-based games are a special case of

concurrent games in which there are optimal strategies that are both stationary and pure.

The *mean* payoff function is considered by Derman in [Derman, 1962] in the context of one player games, *i.e.* Markov decision processes (MDPs). For the mean payoff, every state $s \in \mathbf{S}$ has an immediate reward $r(s) \in \mathbb{R}$, and for a history $s_0 a_0 s_1 a_1 \cdots \in \mathcal{H}$ the payoff is

$$f_{\overline{mean}}(s_0 s_1 \cdots) = \limsup_{n\to\infty} \frac{1}{n+1} \sum_{i=0}^{n} r(s_i).$$

There is also the lim inf version of the payoff function above, denoted by $f_{\underline{mean}}$. Derman proves that for MDPs with the mean payoff function, there exist an optimal strategy that is both stationary and pure. This is done using the fact that Cesàro sum of the sequence of rewards (used in the mean payoff) is equal to the Abel sum (used in the discount payoff), and that the former summability implies the latter. This follows from classical Abel and Tauberian type theorems, see *e.g.* [Hardy, 2000].

First attempt at generalizing the existence of optimal strategies that are both stationary and pure, to two player games was done by Gilette in [Gillette, 1957]. It uses an extension of Littlewood-Hardy theorem [Hardy, 2000] to reduce to the discount payoff. Unfortunately, as pointed by Liggett and Lippman in [Liggett and Lippman, 1969], this extension is false.

Another attempt is made by Liggett and Lippman in the same paper by using a theorem of Blackwell [Blackwell, 1962]. But this proof also contains an error. A counter-example to one of the statements in [Liggett and Lippman, 1969] and an alternate proof is given in the technical report [Gimbert and Kelmendi, 2014a].

In the verification setting, the most useful payoff function is the *parity* payoff. Here every state $s \in \mathbf{S}$ is colored by an integer $c(s) \in \mathbb{N}$, and for a history $s_0 a_0 s_1 a_1 \cdots \in \mathcal{H}$, the payoff is

$$f_{parity}(s_0 s_1 \cdots) = \begin{cases} 1 & \text{if } \limsup_n c(s_n) \text{ is even} \\ 0 & \text{otherwise} \end{cases}.$$

The parity payoff function is important in verification, because for every ω-regular language there is a parity automaton that accepts it, and typically, conditions that we want to verify are given as ω-regular languages. For example, say we are given an ω-regular language $\mathcal{L}$ that describes the correct behavior of the system (in some sense). We use two player stochastic games to model the system: the maximizer is the controller, the minimizer is the environment and we use the parity payoff function, with an appropriate coloring of the states, related to the language $\mathcal{L}$ that is given. The optimal strategy for the maximizer

in such a game is the best controller that the system can have (no matter the behavior of the environment) with respect to fulfilling the correctness specified by $\mathcal{L}$. Using linear programming Courcoubetis and Yannakakis have shown that for MDPs equipped with a parity payoff function, there exist optimal strategies that are both stationary and pure [Courcoubetis and Yannakakis, 1998]. This is done by effectively computing an MDP M' from the given MDP M with the parity payoff, and identifying a set of distinguished states in M' such that "hitting" one of the distinguished states in M' corresponds to fulfilling the parity condition in M. Furthermore, optimal strategies for M' can be transferred to optimal strategies in M. Using linear programming the value[2] of the MDP M' is calculated, and it is shown that M' admits optimal strategies that are stationary and pure.

This result was then generalized to two player stochastic games in [McIver and Morgan, 2002], where the authors build links between an extension of μ-calculus called $qM\mu$ (that has probabilistic choice) and two player stochastic games with the parity payoff. In this quest, they show that both players have optimal strategies that are stationary and pure, thereby proving the positionality of stochastic games equipped with the parity payoff function.

Another proof of this theorem can be found in [Chatterjee *et al.*, 2004]. Here the authors use the fact that there exist *qualitatively optimal* strategies that are both pure and stationary. This is proved in [Chatterjee *et al.*, 2003] by reduction to deterministic parity games.

See also [Zielonka, 2004] for yet another proof of the positionality of two player stochastic parity games. Zielonka gives a straightforward proof by induction on the edges.

An example of stochastic games that are stationary but not as classically studied as the ones above, are one-counter stochastic games. The states are labeled by an element of $\{-1, 0, +1\}$, and the objective is *termination*: the counter reaches value 0. See [Brázdil *et al.*, 2010].

While the results above have very different proofs, there has been some recent work on capturing that which is common among (half) positional games. Central to this effort seem to be two properties: shift-invariance, and being submixing. The former says that the payoff does not change because of a finite prefix of the infinite history. In other words, the infinite history can be *shifted* without a change in the payoff. The latter says that given two histories h_1 and h_2, shuffling them in any way we want into a history h, results in h having a payoff of at most the maximum of the payoffs of h_1 and h_2; *i.e.* we do not gain more by mixing h_1 and h_2. Precise definitions are given in the section that follows.

2 the supremum over the strategies of the expected value of f_{parity}

We start by surveying results on deterministic games. Gimbert and Zielonka in [Gimbert and Zielonka, 2004] use the slightly stronger notion of *fairly mixing* payoff function to prove that deterministic two-player games equipped with such payoff functions admit stationary optimal strategies (players do not have the option of randomization in such deterministic games). This is achieved by induction on the edges, using the same technique as in [Zielonka, 2004]. Moreover the authors point out that classical payoff functions such as parity and mean-payoff are fairly mixing.

This is further improved upon in [Gimbert and Zielonka, 2005]. Here the authors completely characterize the payoff functions for which two-player deterministic games admit stationary optimal strategies. The characterization is done over the preference relations on the set of infinite histories. The sufficient and necessary condition requires that the preference relation and its inverse, is monotone and selective. These two conditions roughly correspond respectively to the shift-invariance and being submixing. As a corollary an interesting property follows: if both players have stationary optimal strategies in one-player games, then they also have them in the two player game.

Progressing to stochastic games, Gimbert in [Gimbert, 2007] proves that for one-player stochastic games (MDPs) if the payoff function is shift-invariant and submixing, then there exist an optimal strategy that is stationary and pure. The proof is by induction on the edges and an application of a zero-one law from probability theory. We will use similar techniques in this chapter.

For two player stochastic games, as for deterministic games, in [Gimbert and Zielonka, 2009] the authors prove that if both players have stationary and pure optimal strategies in one player games, then the two player game is positional as well. This result is, in a sense, furthered by Zielonka in [Zielonka, 2010] by showing that positionality in one player games can be reduced to the existence of optimal and trivial strategies in *multi-armed bandit games*. These are finite sequences of Markov chains, where the decision that the player has to make is which one of the Markov chains to *evolve*. The payoff is a function of the infinite sequence of edges that were taken. A trivial strategy in a multi-armed bandit game is a strategy that chooses to always evolve the same Markov chain. As noted by Zielonka, in practice the submixing and shift-invariant condition is easier to check, but there are one player games that can be proven to admit optimal pure and stationary strategies by reduction to questions on multi-armed bandit games, that do not satisfy shift-invariance or are not submixing.

The results above deal with positionality of two player games. But in many cases we care only for one of the players, while the other represents the adversary, the environment, for whom the possession of simple optimal strategies is not of interest. From here, the question of half-positionality is raised, the question of whether the maximizer has an optimal strategy that is both pure and stationary. Then, if both games equipped with the payoff function f and $-f$ are

half-positional, they are both positional as well. Moreover there are interesting examples in the literature of games that are half-positional but not positional. For example, in the survey article by Grädel [Grädel, 2004], the *Streett-Rabin* game is mentioned as an example of a game that is half-positional but not positional. In a Streett-Rabin game (like in a parity game) the payoff function maps to $\{0,1\}$, a player either wins or not. This winning condition is given by a set $W \subseteq 2^{\mathbf{S}}$ such that the complement of W, $\neg W$, is union-closed; the payoff function maps a history $s_0 s_1 \cdots$ to 1 if and only if the set of states that are seen infinitely often belongs to W.

Another interesting example of games that are half-positional but not positional, are *mean-payoff co-Büchi* games [Chatterjee *et al.*, 2005][3]. The co-Büchi condition is defined by a set $F \subseteq \mathbf{S}$, and it is fulfilled by a history if and only if every state that appears infinitely often belongs to F. The mean-payoff co-Büchi function is a conjunction of the mean payoff and the co-Büchi condition. The idea is that it is of interest (especially when modeling *e.g.* reactive systems) to combine qualitative constraints with quantitative ones.

Going to more general results on half-positional games, Eryk Kopczynski shows in [Kopczyński, 2006] that conditions similar to submixing and shift-invariant imply half-positionality. Moreover, he proves some closure properties and consequently shows families of half-positional games. More details, and more half-positional games that are not positional can be found in [Kopczyński, 2009]. These results are for deterministic two player games, and for payoff functions that take values from $\{0,1\}$.

In this chapter we will report on [Gimbert and Kelmendi, 2014b], where we continue this line of work and prove that in two player stochastic games, if the payoff function is shift-invariant and submixing then the game is half-positional, *i.e.* player Max has an optimal strategy that is stationary and pure. In this way we further generalize [Gimbert, 2007] from one player games to two player games, and [Kopczyński, 2006] from deterministic to stochastic.

The proof is by induction on the set of actions available for Max, using a 0-1 law. It rests on a technical lemma which says that if the payoff function is shift-invariant then both players in the game posses ϵ-subgame-perfect strategies for all $\epsilon > 0$. The proof of this lemma relies on some well-known results from martingale theory, used to analyze the outcome of the game under a reset strategy.

3 in contrast to the half-positionality of mean-payoff co-Büchi games, mean-payoff parity games admit optimal strategies, but they may require infinite-memory

2.2 TWO PLAYER STOCHASTIC GAMES

Let us give precise definitions of two player stochastic games with perfect information, and in doing so, fix the notation.

The game is played between two players. Player 1 is the player whose objective is to maximize the expected payoff, and Player 2 is his enemy. We usually will refer to Player 1 as Max or "the maximizer", whereas Player 2 is referred to as Min or "the minimizer".

If X is a finite set, a probability distribution on X is a function $\delta : X \to [0,1]$ such that for all $x \in X$, $\delta(x) \geq 0$ and $\sum_{x \in X} \delta(x) = 1$. The set of probability distributions over a set X is denoted by $\Delta(X)$.

A game is given as an arena, where the game is played, and a payoff function, for what objective the game is played. The arena was introduced informally in the previous section and illustrated with an example, now we give a precise definition.

Definition 2.1 (Arena).
An arena $\mathcal{A}$ is the tuple $\mathcal{A} = (\mathbf{S_1}, \mathbf{S_2}, \mathbf{A}, (\mathbf{A}(s))_{s \in \mathbf{S}}, p)$, where

- $\mathbf{S_1}$ is a finite set of states controlled by Max,
- $\mathbf{S_2}$ is a finite set of states controlled by Min,
- $\mathbf{A}$ is a finite set of actions,
- for all $s \in \mathbf{S}$, $\mathbf{A}(s) \subseteq \mathbf{A}$ is the set of actions available in state s, and
- $p : \mathbf{S} \times \mathbf{A} \to \Delta(\mathbf{S})$ a function that gives the transition probabilities, or the edges of the arena.

We assume that for all $s \in \mathbf{S}$, $\mathbf{A}(s) \neq \emptyset$, and that the function p is defined only for the pairs (s, a) where $a \in \mathbf{A}(s)$, *i.e.* only for the available actions in s.

We denote by $\mathbf{S}$ the union $\mathbf{S_1} \cup \mathbf{S_2}$.

A few examples of payoff functions were given in the previous section, and it was said that the payoff function is a map from histories to the reals. In reality, in order to make sure that the games we are dealing with have values, we will impose some restrictions that will allow us to use Martin's second determinacy theorem. Since the aim here is to know in which games we have simple optimal strategies, it is favorable to narrow the scope of payoff functions to — at least — the ones that give determined games.

Let X and Y be two nonempty finite sets. By their juxtaposition XY (like $\mathbf{SA}$ in the definition above) we mean the set of all finite sequences (words) xy such that $x \in X$ and $y \in Y$.

We denote by X^* the set of all finite words with alphabet X, including the empty word; by X^+ the set of all finite words with alphabet X excluding the empty word; and by X^ω the set of all infinite words with alphabet X.

Recall that a function $f : (\mathbf{SA})^\omega \to \mathbb{R}$ is Borel-measurable if and only if for all sets X open in $\mathbb{R}$, $f^{-1}(X)$ is Bore (the σ-algebra of Borel sets is the smallest σ-algebra that contains all the open sets), where $(\mathbf{SA})^\omega$ is endowed with the product topology. As a standard reference, see [Kechris, 1995].

Definition 2.2 (Payoff function for an arena).
Let $\mathcal{A}$ be an arena. A payoff function for $\mathcal{A}$ is any function $f : (\mathbf{SA})^\omega \to \mathbb{R}$, that is bounded and Borel-measurable.

A game is defined as a tuple of arena and payoff function.

Definition 2.3 (Two player stochastic game).
A two player stochastic game with perfect information $\mathcal{G}$, is defined as the pair $\mathcal{G} = (\mathcal{A}, f)$ where $\mathcal{A}$ is an arena and f is a payoff function of $\mathcal{A}$.

Let us give the arena of the game in Figure 1 explicitly.

Example 1.
We define explicitly a game $\mathcal{G}$ based in the arena in figure Figure 1 for the mean payoff function. The arena is the tuple $\mathcal{A} = (\mathbf{S_1}, \mathbf{S_2}, \mathbf{A}, (\mathbf{A}(s))_{s\in\mathbf{S}}, p)$, where $\mathbf{S_1} = \{s_0, s_2, s_4\}$ are the states that are controlled by Max, $\mathbf{S_2} = \{s_1, s_3\}$ states controlled by Min. The set of actions is $\mathbf{A} = \{a, \alpha, \beta\}$, and the actions that are available in particular states are given by $\mathbf{A}(s_0) = \mathbf{A}(s_2) = \mathbf{A}(s_4) = \{a\}$, $\mathbf{A}(s_3) = \{\alpha\}$ and $\mathbf{A}(s_1) = \{\alpha, \beta\}$. The function p is defined as $p(s_0, a)(s_1) = p(s_0, a)(s_3) = 1/2$, $p(s_1, \alpha)(s_0) = 1$, etc. We can give the rewards $r(s_i) = i$, $i \in \{0, \ldots, 4\}$, and $f_{\overline{mean}}$ the function that uses them. Then the game $\mathcal{G} = (\mathcal{A}, f_{\overline{mean}})$ is a mean payoff game based on Figure 1. In this game Max does not have any choice, the only action that is enabled in states controlled by him is the action a, so he is obliged to always play it. On the other hand, Min, has a choice in state s_1. If he plays action β then the game will loop in state s_4 and states s_2 and s_3. In the former the mean payoff will be exactly 4, in the latter

the mean payoff will be $5/2$. So playing a β after the first turn will result in a payoff of $\frac{1}{3} \cdot \frac{5}{2} + \frac{2}{3} \cdot 4 = \frac{21}{6}$. But on the other hand if Min always plays α when in state s_1 then eventually we will end up in the loop s_2, s_3 and the amount paid to Max will be $5/2$. Of course this is what Min prefers. △

Usually we will not give examples of games in an explicit manner as above, but rather using only their graphical representation.

2.2.1 *Strategies*

Strategies are the recipes according to which players make their moves. Since we are dealing with perfect-information games, the strategies of the players are functions that, given a finite history, say which action should be played next. The finite history is a finite sequence in $(\mathbf{SA})^*\mathbf{S}$.

Definition 2.4 (Strategies).
A strategy for Max is a function $\sigma : (\mathbf{SA})^*\mathbf{S_1} \to \Delta(\mathbf{A})$, such that for all $h \in (\mathbf{SA})^*$, $s \in \mathbf{S_1}$, $a \in \mathbf{A}$,

$$\sigma(hs)(a) > 0 \implies a \in \mathbf{A}(s).$$

Similarly defined, a strategy for Min is a function $\tau : (\mathbf{SA})^*\mathbf{S_2} \to \Delta(\mathbf{A})$.

As we can see, the type of strategies that we are considering are mixed behavioral strategies, that is to say that at every stage of the game a player can mix between the available actions. For example in state s instead of playing either action a_1 or action a_2, he can *mix* them, by playing a_1 with $1/2$ probability and a_2 with the same probability. Another possibility is to mix strategies themselves. In the beginning of the game, we decide which strategy to follow by flipping a coin, for example. The reason that we do not consider such strategies is that it is a theorem that they are not more powerful than behavioral strategies, in perfect information games. This result is known as *Kuhn's theorem*. This stops being true without the perfect information hypothesis. See the example of the absent-minded driver in [Piccione and Rubinstein, 1997].

We follow standard notation, by denoting with σ strategies for Max, and by τ strategies for Min, with possible subscripts or superscripts.

As mentioned before, we are interested in a class of particularly simple strategies: the pure and stationary strategies.

Definition 2.5 (Pure and stationary strategies).
Let σ be a strategy for Max. We say that σ is *pure* if for all finite histories $h = s_0 \cdots s_n \in (\mathbf{SA})^* \mathbf{S_1}$ there exists $a \in \mathbf{A}(s_n)$ such that $\sigma(h)(a) = 1$.

We say that σ is *stationary* if for all finite histories $h = s_0 \cdots s_n \in (\mathbf{SA})^* \mathbf{S_1}$, $\sigma(h) = \sigma(s_n)$.

Similarly, one defines stationary and pure strategies for Min.

So a strategy is pure if at every stage of the game, it chooses deterministically a single action, *i.e.* it always maps to Dirac distributions on $\mathbf{A}$, and it is stationary if it is solely a function of the current state, and not of the previous states and actions played in the game. In particular we see that strategies that are both stationary and pure are functions $\sigma : \mathbf{S_1} \to \mathbf{A}$.

2.2.2 *Probability measure*

In order to talk about the probability of certain events happening, or the *expected* payoff we have to define a probability space. Indeed, once an initial state $s_0 \in \mathbf{S}$ is fixed, together with strategies σ, τ belonging to Max and Min respectively, we can construct a probability space whose sample space is $(\mathbf{SA})^\omega$, the set of infinite histories. The σ-algebra, is the one that is generated by the cylinders $h(\mathbf{SA})^\omega$ and $h'(\mathbf{AS})^\omega$, where $h \in (\mathbf{SA})^*$ and $h' \in (\mathbf{SA})^*\mathbf{S}$. So a cylinder $h(\mathbf{SA})^\omega$ is the set of infinite words whose prefix is h.

In this σ-algebra there is a unique probability measure, which we denote by $\mathbb{P}_{s_0}^{\sigma,\tau}$, such that the following holds for all $h = r_0 \cdots r_n \in (\mathbf{SA})^*\mathbf{S}$, $t \in \mathbf{S}$ and $a \in at$,

$$\mathbb{P}_{s_0}^{\sigma,\tau}[ha \mid h] = \begin{cases} \sigma(h)(a) & \text{if } r_n \in \mathbf{S_1}, \\ \tau(h)(a) & \text{if } r_n \in \mathbf{S_2}, \end{cases}$$
$$\mathbb{P}_{s_0}^{\sigma,\tau}[hat \mid h] = p(r_n, a)(t).$$

Above, to simplify the notation, for the cylinder $ha(\mathbf{SA})^\omega$, we wrote $\mathbb{P}_{s_0}^{\sigma,\tau}[ha]$ instead of $\mathbb{P}_{s_0}^{\sigma,\tau}[ha(\mathbf{SA})^\omega]$, and we will continue to do so in the sequel.

The notation in the left hand sides of the two equations above are standard conditional probabilities. The first equation says that the probability measure follows the two strategies σ and τ, and the second equation says that it follows the transition probabilities of the arena.

The degenerate case of $\mathbb{P}_{s_0}^{\sigma,\tau}[s]$ for $s \in \mathbf{S}$ is defined to be equal to 1 if $s = s_0$, otherwise it is equal to 0; so that the probability measure also follows the initial state.

We denote by $\mathbb{E}_{s_0}^{\sigma,\tau}$, the expectation corresponding to the probability measure above.

In order to make precise what we mean when we say the expected payoff, or the expected value of a function let us define a family of random variables that will be used throughout this chapter. For all $n \in \mathbb{N}$ define the random variables S_n, A_n to be such that for all outcomes $s_0 a_0 s_1 a_1 \cdots \in (\mathbf{SA})^\omega$, $S_n(s_0 a_0 s_1 \cdots) = s_n$ and $A_n(s_0 a_0 s_1 \cdots) = a_n$.

The expected value of a random variable is defined as usual, and the expected value of a function f whose domain is the sample space $(\mathbf{SA})^\omega$, denoted by $\mathbb{E}_{s_0}^{\sigma,\tau}[f]$, is just the expectation of the random variable $f(S_0 A_0 S_1 \cdots)$.

2.2.3 *Values and optimal strategies*

Let $\mathcal{G}$ be a game with payoff function f; by definition f is bounded and Borel-measurable. The expected payoff when starting from state $s_0 \in \mathbf{S}$ and players play with the respective strategies σ and τ is $\mathbb{E}_{s_0}^{\sigma,\tau}[f]$. This is what Max wants to maximize (and Min to minimize). Let τ be some strategy for Min, then the *best response* to τ is some strategy σ_τ for Max such that for all other strategies σ, $\mathbb{E}_{s_0}^{\sigma_\tau,\tau}[f] \geq \mathbb{E}_{s_0}^{\sigma,\tau}[f]$. The expected payoff under the best response for τ is the most that Max can win, given that Min has chosen τ.

The best response does not always exist, but in any case it can be approximated, by better and better responses converging to $\sup_\sigma \mathbb{E}_{s_0}^{\sigma,\tau}[f]$. Min, true to his name, will want to minimize, hence $\inf_\tau \sup_\sigma \mathbb{E}_{s_0}^{\sigma,\tau}[f]$, called the *minmax* of the game $\mathcal{G}$ is of interest. Symmetrically the maxmin of the game is defined to be $\sup_\sigma \inf_\tau \mathbb{E}_{s_0}^{\sigma,\tau}[f]$.

Naturally knowing what strategy your opponent will choose before choosing your own is advantageous, so it is easy to prove that

$$\inf_\tau \sup_\sigma \mathbb{E}_{s_0}^{\sigma,\tau}[f] \geq \sup_\sigma \inf_\tau \mathbb{E}_{s_0}^{\sigma,\tau}[f].$$

In many cases the inverse inequality holds as well, *i.e.* , minmax is equal to maxmin providing an equilibrium point. Such minimax theorems are central in game theory, in fact, the proof of the minimax theorem on matrix games by Von Neumann is the genesis of game theory.

In our case minmax is equal to maxmin as well, this is a corollary of another celebrated result.

Theorem 2.6 (Martin's second determinacy theorem [Martin, 1998]). Let $\mathcal{G}$ be a game with payoff function f (so f is Borel-measurable and bounded) then

$$\inf_{\tau}\sup_{\sigma}\mathbb{E}_{s_0}^{\sigma,\tau}[f] = \sup_{\sigma}\inf_{\tau}\mathbb{E}_{s_0}^{\sigma,\tau}[f].$$

This quantity is called the value of the game.

Definition 2.7 (Value of the game).
Let $\mathcal{G}$ be a game with payoff function f and $s \in \mathbf{S}$. The value of the state s is defined as

$$val(\mathcal{G}, s) = \inf_{\tau}\sup_{\sigma}\mathbb{E}_{s_0}^{\sigma,\tau}[f] = \sup_{\sigma}\inf_{\tau}\mathbb{E}_{s_0}^{\sigma,\tau}[f].$$

When the game $\mathcal{G}$ is clear from the context, we omit it and write only $val(s)$.

Intuitively the value of the game is the quantity that either of the players can win at most, given that their enemy plays his best strategy. So it is related to the notion of optimal strategies.

Definition 2.8 (ϵ-optimal strategies).
Let $\epsilon > 0$ and σ a strategy for Max. We say that σ is *ϵ-optimal* if and only if for all states $s \in \mathbf{S}$ and all strategies τ for Min,

$$\mathbb{E}_{s}^{\sigma,\tau}[f] \geq val(s) - \epsilon.$$

Symmetrically define ϵ-optimal strategies for Min. We say that a strategy is *optimal* if it is 0-optimal.

Note the difference between the best response and an optimal strategy. If Max has an optimal strategy $\sigma^{\#}$, and Min plays with a particularly bad strategy τ, then $\sigma^{\#}$ in general is not the best response to τ.

2.2.4 *A Half-positional example*

Let us consider a Streett-Rabin game.

Example 2.
Let $\neg F$ be the closure under union of $\{\{A, a\}, \{B, b\}, \{A, B, a, b\}, \{D\}\}$. For an infinite history $h \in (\mathbf{SA})^\omega$ define $Inf(h)$ to be the set of states that are seen infinitely often. The Rabin-Strett payoff function f_{rs} is defined as

$$f_{rs}(h) = \begin{cases} 1 & \text{if } Inf(h) \in F \\ 0 & \text{otherwise.} \end{cases}$$

Let $\mathcal{G}$ be the game with the arena depicted in the following figure and payoff f_{rs}.

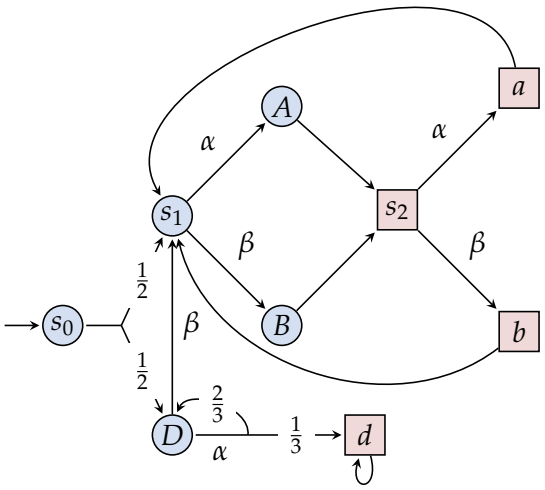


Figure 2: A Streett-Rabin game

Player Min prefers that the set of states that are seen infinitely often to be in $\neg F$. Moreover observe that $\neg F$ is by definition closed under union.

Consider the strategy for Min that plays as follows. When the game is in state s_2, if in the previous turn it was in A play α, if it was in B play β. This ensures that whatever Max plays in the upper part of the game, the set of states that are seen infinitely often will not be an element of F. So in this part of the game Min can win, no matter the strategy of Max. Therefore we can conclude that $val(s_1) = 0$ and the same for the values of A, a, B, b and s_2; since this is a strongly connected component of the arena.

In state D, Max has the choice between playing β and going to the strongly connected component above, where Min wins, or repeatedly playing α and almost surely landing on the state d. Of course the latter is more preferable, especially since $\{d\} \in F$. In this part of the arena Min does not control anything (state d can only self-loop) so we can conclude that $val(D) = val(d) = 1$. Consequently, since s_0 has a single action that takes the game to either state s_1 or D with equal probability, and it has no incoming edge, we conclude that $val(s_0) = 1/2$.

A stationary and pure strategy that is optimal for Max is the strategy σ defined as $\sigma(s_1) = \sigma(D) = \alpha$, *i.e.* always play action α. With this strategy Max makes sure that the payoff is at least $1/2$. Min has a simple strategy as a best response to σ — the stationary and pure strategy τ that simply plays $\tau(s_2) = \alpha$. Nevertheless τ is not an optimal strategy; Max can beat it with a strategy σ' that plays $\sigma'(s_1) = \beta$, getting a payoff 1.

We observe that while Max indeed has an optimal strategy that is pure and stationary, that is not the case for Min. In order for Min to win in the upper part of the arena, he needs to remember the last state seen before making his decision. So this is an example of a half-positional game that is not positional.△

In the example above we see that there might be more than one optimal strategy that is stationary and pure, and that such a strategy is not always the best response, but it makes sure that the player that is playing it, receives at least the value as payoff.

2.2.5 *Martingales*

Consider the stochastic process (sequence of random variables) $(val(S_n))_{n\in\mathbb{N}}$, if Max plays only optimal actions then this process on average increases, a stochastic process that increases in this way is called a *supermartingale*. Martingales are a basic notion in probability theory of a type of a stochastic process whose expected value, knowing everything up to a certain date, is equal to the value of the process at that same date. In this section we will recall the definition of martingales and prove a lemma which we will use later.

Definition 2.9 (Martingale and stopping time).
In a filtered probability space with some filtration $(F_n)_{n\in\mathbb{N}}$ a (discrete-time) *martingale* is a stochastic process $(X_n)_{n\in\mathbb{N}}$ such that for all $n \in \mathbb{N}$,

$$\mathbb{E}[|X_n|] < \infty,$$
$$\mathbb{E}[X_{n+1} \mid F_1, \ldots, F_n] = X_n.$$

A *stopping time* with respect to a filtration $(F_n)_{n\in\mathbb{N}}$ is a random variable T taking values in $\mathbb{N} \cup \{\infty\}$ such that for all $t \in \mathbb{N}$ the event $T \leq t$ is in F_t.

To get the definition of a *supermartingale* and *submartingale* replace in the definition above the equality in the second equation by $\geq$ and $\leq$ respectively.

The following two theorems can be found in any textbook on probability or measure theory, see *e.g.* [Williams, 1991].

Theorem 2.10 (Lebesgue's dominated convergence theorem).
Let $(f_n)_{n\in\mathbb{N}}$ be a sequence of real-valued measurable functions with respect to the measure space (S, Σ, μ). Assume that (f_n) converges point-wise to some function f and that there exists an integrable function g such that for all $n \in \mathbb{N}$ and $x \in S$, $|f_n(x)| \leq g(x)$, then f is integrable as well and

$$\lim_{n\to\infty} \int_S f_n \, d\mu = \int_S f \, d\mu.$$

Above by a function f is *integrable* we mean $\int_S |f| \, d\mu < \infty$.

Theorem 2.11 (Doob's martingale convergence theorem).
Let $(X_n)_{n\in\mathbb{N}}$ be a martingale. Suppose that there exists some $K \in \mathbb{R}$ such that for all h in the sample space and $n \in \mathbb{N}$, $|X_n(h)| \leq K$, then the point-wise limit

$$X(h) = \lim_{n\to\infty} X_n(h)$$

exists and is finite for all outcomes h.

This theorem also holds for sub and supermartingales.

For the sake of clarity, the theorems above, are weaker versions that are easier to state, but sufficient for our purposes.

The lemma that follows deals with a martingale whose value we want to know at some time T that depends on the outcome, *i.e.* it is a stopping time.

Lemma 2.12.
Let T be a stopping time with respect to the natural filtration induced by the sequence $(S_n)_{n\in\mathbb{N}}$, and $(X_n)_{n\in\mathbb{N}}$ a martingale. Assume that there exists some $K \in \mathbb{R}$ such that for all h in the sample space and $n \in \mathbb{N}$, $|X_n(h)| \leq K$. We define the random variable X_T as

$$X_T = \begin{cases} X_n & \text{if } T \text{ is finite and equal to } n, \\ \lim_{n\to\infty} X_n & \text{if } T = \infty. \end{cases}$$

Then

$$\mathbb{E}[X_T] = \mathbb{E}[X_0].$$

Proof.
First we show that X_T is well-defined. Indeed this follows immediately from Theorem 2.11, *i.e.* the limit $\lim_{n\to\infty} X_n$ exists in the definition of X_T.

We define another process: the stopped process $(Y_n)_{n\in\mathbb{N}}$ as

$$Y_n = X_{\min(T,n)},$$

for all $n \in \mathbb{N}$. An immediate observation is that $(Y_n)_{n\in\mathbb{N}}$ is a martingale as well, and that it has the bound K. Applying Theorem 2.11 to $(Y_n)_{n\in\mathbb{N}}$, we conclude that it converges. From the definition of X_T, we have that $\lim_{n\to\infty} Y_n = X_T$. Now applying Theorem 2.10 to $(Y_n)_{n\in\mathbb{N}}$ and X_T, we have

$$\lim_{n\to\infty} \mathbb{E}[Y_n] = \mathbb{E}[X_T]. \tag{1}$$

Since $(Y_n)_{n\in\mathbb{N}}$ is a martingale we can prove that $\mathbb{E}[Y_n] = \mathbb{E}[Y_0] = \mathbb{E}[X_0]$. This together with (1) concludes the proof. ∎

If we replace martingale by supermartingale or submartingale, the lemma above still holds with the equality replaced by the appropriate inequality.

2.3 SHIFT-INVARIANT AND SUBMIXING PAYOFF FUNCTIONS

We formally introduced what a payoff function for an arena is in Definition 2.2. In this section we will give the definitions of shift-invariance and being submixing, the two conditions on payoff functions whose conjunction is sufficient for half-positionality.

In order to simplify the notation we decouple the payoff function from the arena in the following way. Let $\mathbf{C}$ be a set of colors. Any bounded and Borel-measurable function $f : \mathbf{C}^\omega \to \mathbb{R}$ is called a payoff function. Given an arena $\mathcal{A}$, we color it by a function $c : \mathbf{SA} \to \mathbf{C}$, and extend the coloring to infinite histories $\bar{c} : (\mathbf{SA})^\omega \to \mathbf{C}^\omega$ in the obvious way. We equip the arena $\mathcal{A}$ with the payoff function f' in $\mathcal{A}$ to form the game $\mathcal{G} = (\mathcal{A}, f')$ where $f' = f \circ \bar{c}$.

A shift-invariant payoff function is one that does not change when adding some finite prefix.

Definition 2.13 (Shift-invariant payoff function).
A payoff function $f : \mathbf{C}^\omega \to \mathbb{R}$ is *shift-invariant* if for all $p \in \mathbf{C}^*$ and $h \in \mathbf{C}^\omega$,

$$f(ph) = f(h).$$

A similar (but sometimes not equivalent) property is found in the literature under the name of *tail-measurability*. All the payoff functions that we have seen before, except the discount payoff, are shift-invariant. This is fairly trivial to prove, as is usually the case for shift-invariance.

The submixing property roughly says that given $c_1, c_2 \in \mathbf{C}^\omega$, we cannot gain a larger payoff by shuffling c_1 and c_2. We make this formal.

Let $h \in \mathbf{C}^\omega$, a *factorization* of h is a sequence of finite words $w_1, w_2, \ldots \in \mathbf{C}^*$, such that $h = w_1 w_2 \cdots$.

Definition 2.14 (Shuffle).
Let $h, h_1, h_2 \in \mathbf{C}^\omega$. We say that h is a *shuffle* of h_1 and h_2, if there exist factorizations $w_1, w_2, \ldots$ and $v_1, v_2, \ldots$ of h_1 and h_2 respectively such that

$$h = w_1 v_1 w_2 v_2 \cdots.$$

Now we can define the submixing property.

Definition 2.15 (Submixing payoff function).
A payoff function f is said to be *submixing* if for all $h, h_1, h_2 \in \mathbf{C}^\omega$ such that h is a shuffle of h_1 and h_2,

$$f(h) \leq \max(f(h_1), f(h_2)).$$

The main theorem of this chapter says that if the payoff function of the game is both shift-invariant and submixing then the game is half-positional. Using this theorem we will be able to recover a few of the classical results on positionality and further provide examples of games that are half-positional.

Example 3.
Consider the Streett-Rabin payoff function f_{rs} induced by the winning set $F \subset 2^{\mathbf{S}}$. Recall that by definition $\neg F$ is closed under union and the payoff function is given as

$$f_{rs}(h) = \begin{cases} 1 & \text{if } Inf(h) \in F \\ 0 & \text{otherwise.} \end{cases}$$

We can imagine that the set of colors is $\mathbf{S}$ and the domain of f_{rs} is $\mathbf{S}^\omega$. Since the payoff is defined in terms of Inf, the set of states that are seen infinitely often we see immediately that it is shift-invariant.

In order to show that it is submixing as well, let $h, h_1, h_2 \in \mathbf{S}^\omega$ be such that h is a shuffle of h_1 and h_2. With the purpose of reaching a contradiction assume that $f(h) = 1$ and $f(h_1) = f(h_2) = 0$. From the definition it follows that $Inf(h_1) \in \neg F$, and $Inf(h_2) \in \neg F$ but $Inf(h) = Inf(h_1) \cup Inf(h_2) \in F$; this contradicts the hypothesis that $\neg F$ is closed under union. △

2.4 Existence of ϵ-subgame-perfect Equilibria

In this section we will prove a technical lemma that has to do with existence of ϵ-subgame-perfect strategies. These are strategies that are not only ϵ-optimal from the beginning of the play, but for any state that is visited throughout the history, they remain ϵ-optimal with respect to that state. That is to say that, when playing with an ϵ-subgame-perfect strategy it never happens that when we reach some state of the game that is more lucrative than the state from which we have started, and we do not take advantage.

In a sense, ϵ-subgame-perfect strategies are ϵ-optimal strategies that take advantage of non-optimal decisions that the enemy makes, in this way they are strategies that are ϵ-optimal but also get close to being best responses.

We first introduce the notion of shifted function with domain $(\mathbf{SA})^\omega$. Let $h = s_0 \cdots s_n \in (\mathbf{SA})^*\mathbf{S}$ be a finite history and f some function whose domain is the set of infinite histories (the sample space) $(\mathbf{SA})^\omega$. The function f *shifted* by the finite history h, denoted by $f[h]$ is defined by

$$f[h](t_0 b_0 t_1 b_1 \cdots) = \begin{cases} f(s_0 \cdots t_0 b_0 \cdots) & \text{if } s_n = t_0, \\ f(t_0 b_0 \cdots) & \text{otherwise.} \end{cases}$$

So $f[h]$ is the same as f except that it inserts h as the prefix of its parameter.

Definition 2.16 (ϵ-subgame-perfect strategy).
Let σ be a strategy for Max and $\epsilon > 0$. We say that σ is ϵ-subgame-perfect if for all $h = s_0 \cdots s_n \in \mathbf{S}(\mathbf{AS})^*$,

$$\inf_\tau \mathbb{E}_{s_n}^{\sigma[h],\tau}[f[h]] \geq val(s_n) - \epsilon.$$

ϵ-subgame-perfect strategies for Min are defined symmetrically.

In this section we will prove the following lemma.

Lemma 2.17 ([Gimbert and Kelmendi, 2014b; Mashiah-Yaakovi, 2014]).
Let f be a shift-invariant payoff function, in a game $\mathcal{G}$. Then for all $\epsilon > 0$ both players have ϵ-subgame-perfect strategies in $\mathcal{G}$.

In parallel to our work, a simpler proof of Lemma 2.17 was given in [Mashiah-Yaakovi, 2014]. While it is stated that there are ϵ-subgame-perfect strategies for all payoff functions that are Borel-measurable and bounded; the notion of ϵ-subgame-perfect strategy used in [Mashiah-Yaakovi, 2014] is slightly different from the one we have here. Indeed there need not exist ϵ-subgame-perfect strategies if the payoff function is not shift-invariant as seen in the following example.

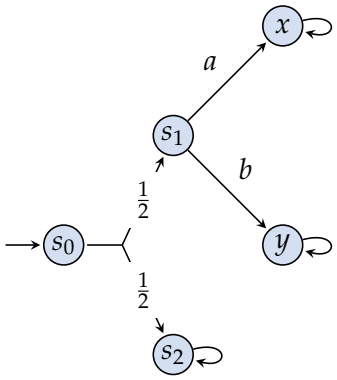


Figure 3: A game with no ϵ-subgame-perfect strategies

Example 4.
Consider the one player arena in Figure 3 equipped with the payoff function f that is not shift-invariant, defined as follows. It takes as input $\mathbf{S}^\omega$ where $\mathbf{S} = \{s_0, s_1, s_2, x, y\}$ and maps to $\{0, \frac{1}{2}, 1\}$. It returns $\frac{1}{2}$ for the sequence $s_0 s_1 x^\omega$ and 1 for the sequence $s_1 y^\omega$, for all other inputs it returns 0. We see that $val(s_0) = \frac{1}{4}$ with the optimal strategy that plays a. Similarly $val(s_1) = 1$ with the strategy that plays b. But no ϵ-subgame-perfect strategy exists with $0 < \epsilon < 1/2$. To see this, consider the history $h = s_0 a s_1$. An ϵ-subgame-perfect strategy σ ensures

$$\mathbb{E}_{s_1}^{\sigma[h]}[f[h]] \geq val(s_1) - \epsilon > \frac{1}{2},$$

which implies that $\mathbb{E}_{s_0}^{\sigma}[f] > \frac{1}{4}$. This is not possible since $val(s_0) = \frac{1}{4}$. △

Before moving on to the technical details of the proof let us give a rough idea of how it goes.

Proof Idea (Lemma 2.17).
Martin's determinacy theorem (Theorem 2.6) implies that there exist ϵ-optimal strategies, but a strategy that is ϵ-optimal need not be ϵ-subgame-perfect. In particular imagine a game with payoff function whose lower bound is b and s some state with $val(s) = b$. Here Max can choose any strategy to be $\sigma[s]$ and still remain optimal.

We will take an ϵ-optimal strategy σ, and detect when playing with σ is not ϵ-subgame-perfect, *i.e.* when there exist some $h = s_0 \cdots s_n$ such that $\sigma[h]$ is not ϵ-optimal from s_n. We will call this a *weakness*, and simply reset the memory whenever we detect it. We do this because σ is ϵ-optimal from s_n even though $\sigma[h]$ might not be.

If the number of memory resets that we have to perform is finite then the reset strategy is ϵ-subgame-perfect and we are done. What then remains to show is that almost surely the number of resets that is performed is finite. This

can be shown by choosing particular ϵ-optimal strategies, namely ones that only play actions that preserve the value. △

Let $\mathcal{G} = (\mathcal{A}, f)$ be some game whose payoff function is shift-invariant. For the sake of clarity we fix here for the rest of this section some $\epsilon > 0$ and prove that there exists 2ϵ-subgame-perfect strategies. We will do the proof for Max, but symmetrically one can prove the lemma for Min as well.

2.4.1 *Weaknesses and the reset strategy*

Let σ be a strategy for Max. If after some finite history σ is not 2ϵ-optimal in the subgame, we say that the history is a σ-weakness.

Definition 2.18 (σ-weakness).
Let σ be a strategy for Max. A finite history $h \in \mathbf{S}(\mathbf{AS})^*$ is a *σ-weakness* if $\sigma[h]$ is not 2ϵ-optimal.

Observe that if some strategy σ has no σ-weaknesses then it is 2ϵ-subgame-perfect. Given a strategy σ and $h \in (\mathbf{SA})^\omega$, we say that a *weakness occurs in h* if there exists some finite prefix of h that is a σ-weakness.

We will factorize $h \in (\mathbf{SA})^\omega$ according to its σ-weaknesses in the following way. Let $h = s_0 a_0 \cdots$, and take $h_0 = s_0 a_0 s_1 \cdots s_n$ to be the shortest prefix of h that is a σ-weakness. Take h_1 to be the shortest prefix of $s_n a_n \cdots$ that is a σ-weakness and so on. In this way we construct a possibly infinite sequence of finite histories[4] $h_0, h_1, \ldots$ such that for all $n \in \mathbb{N}$ for which h_n exists we have

- if h_n is finite then it is a σ-weakness,
- if h_n is finite then no strict prefix of h_n is a σ-weakness,
- if h_n is infinite then no prefix of h_n is a σ-weakness and $h = h_0 \cdots h_n$.

The reset strategy will reset its memory whenever it detects a weakness, in particular it will reset it after reading each of the factors $h_0, h_1, \ldots$.

4 if the sequence is finite, then the last element is an infinite history

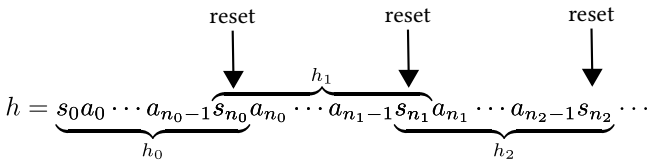


Figure 4: Factorizing an infinite history into weaknesses

Example 5.
Consider the deterministic game with the arena in Figure 5.

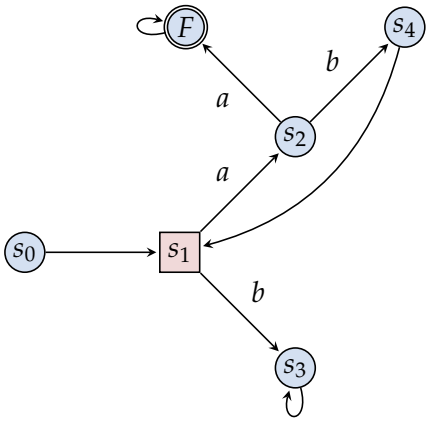


Figure 5: A Büchi game

The payoff is the Büchi payoff, which is a special case of f_{parity}; Max wins if and only if the state F is seen infinitely often. We observe that the value of s_2 and F is 1 and the value of every other state is 0. Consider the strategy σ for Max that plays as follows. If we have seen s_0 in the past when we are in state s_2 play b, otherwise play a. While σ is not a particularly intelligent strategy, it is optimal. When the game starts from s_0 and Min plays a in s_1, σ does not take advantage of this mistake, but in any case the value of s_0 is 0 so σ remains optimal. But it is not ϵ-subgame-perfect, for any $\epsilon < 1$. In fact the finite history $s_0 a s_1 a s_2$ is a σ-weakness. If at this stage we reset the memory of σ, it will not remember that the state s_0 has been seen and play as if the game has started from s_2 and consequently make the good choice of playing a. This induces the infinite history $s_0 a s_1 a s_2 (aF)^\omega$, which is factorized as above into $h_0 = s_0 a s_1 s_2$ and $h_1 = s_2 (aF)^\omega$. △

We give the definition of the *date of weakness*, which serves for the presentation of the proofs that will follow.

Definition 2.19 (Date of weakness).
Let σ be a strategy for Max and $s_0 \cdots s_n \in \mathbf{S}(\mathbf{AS})^*$ a finite history. Define inductively on n the function

$$\delta_\sigma(s_0 \cdots s_n) = \begin{cases} n & \text{if } h \text{ is a } \sigma\text{-weakness,} \\ \delta_\sigma(s_0 \cdots s_{n-1}) & \text{otherwise,} \end{cases}$$

where $\delta_\sigma(s_0) = 0$ and $h = s_{\delta_\sigma(s_0 \cdots s_{n-1})} \cdots s_n$.

We say that $\mathcal{W}_\sigma(s_0 \cdots s_n)$ is true if and only if $\delta_\sigma(s_0 \cdots s_n) = n$. $\mathcal{W}_\sigma(s_0 \cdots s_n)$ can be read as $s_0 \cdots s_n$ *is a* σ*-weakness.*

In particular, with regard to the factorization of $h = s_0 a_0 \cdots$ into $h_0, h_1, \cdots$, the function δ_σ, returns the dates $n_0, n_1, n_2, \ldots$. *i.e.* for any strict prefix of h_0 it returns 0, for h_0 itself and every strict prefix of $h_0 h_1$ returns n_0, for $h_0 h_1$ itself returns n_1 and so on. Intuitively δ_σ gives the date of the last σ-weakness in the finite history. Let us use it to define the *reset strategy*.

Definition 2.20 (Reset strategy).
Let σ be a strategy for Max. We define $\hat{\sigma}$, *the reset strategy based on* σ as

$$\hat{\sigma}(s_0 \cdots s_n) = \sigma(s_{\delta_\sigma(s_0 \cdots s_n)} \cdots s_n).$$

So the strategy $\hat{\sigma}$ plays like the strategy σ except that it ignores the part of the history that comes before the last σ-weakness.

2.4.2 *Locally-optimal strategies*

We will base the reset strategy in a particular class of ϵ-optimal strategies, the *locally-optimal* strategies. These are strategies that at every stage make decisions that do not decrease the value (from Max's point of view). The reason why we consider this subclass of strategies is that there exist ϵ-optimal strategies (that are not locally-optimal) for which the reset strategy has to reset infinitely many times with positive probability. This is not the case for locally-optimal strategies.

Definition 2.21 (Value-preserving action and locally-optimal strategy). An action $a \in \mathbf{A}(s)$ is *value-preserving* (in s) if and only if

$$val(s) = \sum_{t \in \mathbf{S}} p(s,a)(t)val(t).$$

In case it is not value-preserving we say that it is *value-changing*.

A strategy that plays only value-preserving actions is called *locally-optimal*.

Observe that a locally-optimal strategy is not necessarily optimal. Imagine a one-player game with two states s_0 and s_1 with action a in s_0 we loop, and with action b we go to state s_1. The game is won if and only if we see the state s_1 infinitely often. Both states have value 1 and in s_0 both actions are value-preserving, so the strategy that keeps playing a and looping in state $s_0$is locally-optimal but not optimal.

On the other hand, an optimal strategy must be locally-optimal, otherwise the opponent can take advantage of the mistake and get a larger payoff.

When a player plays with a locally-optimal strategy, he imposes some regularity on the stochastic process $(val(S_n))_{n\in\mathbb{N}}$. In particular if Max (Min) plays with a locally-optimal strategy, then the process $(val(S_n))_{n\in\mathbb{N}}$ is a supermartingale (submartingale). If both Max and Min play with a locally-optimal strategy then the process is a martingale.

Proposition 2.22.
Let σ,τ be strategies for Max and Min and $s_0 \in \mathbf{S}$. If σ is locally-optimal then $(val(S_n))_{n\in\mathbb{N}}$ is a supermartingale, for the probability measure $\mathbb{P}_{s_0}^{\sigma,\tau}$. If τ is locally-optimal then $(val(S_n))_{n\in\mathbb{N}}$ is a submartingale.

Proof.
We will consider the probability space that is filtered by the filtration induced by the process $(S_n)_{n\in\mathbb{N}}$. Since the payoff function f is bounded, the expected value of S_n is bounded as well. From the definitions we have that the expected value at date $n+1$,

$$\mathbb{E}_{s_0}^{\sigma,\tau}[val(S_{n+1}) \mid S_0, \ldots, S_n]$$

is equal to

$$\begin{cases} \sum_{a\in\mathbf{A}(S_n)} \sigma(S_0\cdots S_n)(a)\big(\sum_{t\in\mathbf{S}} p(S_n,a)(t)\cdot val(t)\big) & \text{if } S_n\in\mathbf{S_1}, \\ \sum_{a\in\mathbf{A}(S_n)} \tau(S_0\cdots S_n)(a)\big(\sum_{t\in\mathbf{S}} p(S_n,a)(t)\cdot val(t)\big) & \text{if } S_n\in\mathbf{S_2}. \end{cases}$$

Since σ is locally-optimal, it follows that

$$\mathbb{E}_{s_0}^{\sigma,\tau}[val(S_{n+1}) \mid S_0,\dots,S_n] \geq val(S_n).$$

The process $(val(S_n))_{n\in\mathbb{N}}$ is a supermartingale. Symmetrically we can prove that if τ is locally-optimal, the process of values is a submartingale; if both σ and τ are locally-optimal then $(val(S_n))_{n\in\mathbb{N}}$ is a martingale. ∎

We continue proving properties of the game when Max plays with a locally-optimal strategy. One interesting property, is that if one of the players plays with a locally-optimal strategy, he eventually forces the opponent to do the same. While this might seem counter-intuitive in a sense, because playing with a good strategy is forcing the opponent to also play with a somewhat good strategy, it becomes clearer if we think of it as follows. Suppose that it is Max that plays with a locally-optimal strategy. At each stage that Min plays a value-changing action, he damages his payoff, and since Max is playing with locally-optimal strategy he takes advantage of this. But it is not possible to keep taking advantage and increasing the payoff for Max and decreasing it for Min since the payoff function is bounded and we have a finite arena.

In fact we will prove a stronger property: if Max plays with a locally-optimal strategy then both players will eventually play only *stable actions*.

Definition 2.23 (Stable actions).
Let $s\in\mathbf{S}$ and $a\in\mathbf{A}(s)$. We say that a is stable (in s) if for all $t\in\mathbf{S}$,

$$p(s,a)(t) > 0 \implies val(s) = val(t)$$

A stable action is necessarily value-preserving. On the other hand a value-changing action cannot be stable. So being stable is strictly stronger than being value-preserving.

Lemma 2.24.
Let σ and τ be strategies for Max and Min respectively, and $s_0\in\mathbf{S}$. If σ is locally-optimal then

$$\mathbb{P}_{s_0}^{\sigma,\tau}[\{\exists n, \forall k\geq n, A_n \text{ is stable}\}] = 1.$$

Proof.
In case every action is stable, we are done, therefore assume that there exists $s \in \mathbf{S}$ and $a \in \mathbf{A}(s)$ such that a is not stable in s. Let $t \in \mathbf{S}$ be such that $p(s,a)(t) > 0$ and $val(s) \neq val(t)$. Denote the event "we see action a infinitely often" by $\mathcal{E}_{sa}$ defined formally as

$$\mathcal{E}_{sa} = \{\forall m, \exists n \geq m, S_n = s \wedge A_n = a\}.$$

We will prove that $\mathbb{P}_{s_0}^{\sigma,\tau}[\mathcal{E}_{sa}] = 0$ by contradiction. Assume that $\mathbb{P}_{s_0}^{\sigma,\tau}[\mathcal{E}_{sa}] > 0$. It follows that $\mathbb{P}_{s_0}^{\sigma,\tau}[\mathcal{E}_{sat}] > 0$ where $\mathcal{E}_{sat}$ is the event defined as

$$\mathcal{E}_{sat} = \{\forall m, \exists n \geq m, S_n = s \wedge A_n = a \wedge S_{n+1} = t\}.$$

From here we conclude that there is nonzero probability that for infinitely many $n \in \mathbb{N}$,

$$|val(S_n) - val(S_{n+1})| \geq val(t) - val(s) > 0.$$

In other words, there is some nonzero probability that the process $(val(S_n))_{n\in\mathbb{N}}$ does not converge. But having σ locally-optimal, from Proposition 2.22 the process $(val(S_n))_{n\in\mathbb{N}}$ is a supermartingale, and Doob's theorem (Theorem 2.11) implies that it converges almost surely. Hence the assumption that $\mathbb{P}_{s_0}^{\sigma,\tau}[\mathcal{E}_{sa}] > 0$ must be false. ∎

By weakening the stability of the action to it being value-preserving we get the following corollary.

Corollary 2.25.
Let σ and τ be strategies for Max and Min respectively, and $s_0 \in \mathbf{S}$. If σ is locally-optimal then

$$\mathbb{P}_{s_0}^{\sigma,\tau}[\{\exists n, \forall k \geq n, A_n \text{ is value-preserving}\}] = 1.$$

The proof is completely symmetric if we suppose that τ is locally-optimal.

Another useful property of the game when Max plays with a locally-optimal strategy is that if we stop the process $(val(S_n))_{n\in\mathbb{N}}$ at some stopping time, the value will be larger then the value of the initial state.

Lemma 2.26.
Let T be a stopping time with respect to the filtration induced by $(S_n)_{n\in\mathbb{N}}$, σ and τ strategies for Max and Min respectively and $s_0 \in \mathbf{S}$. If σ is locally-optimal then the process $(val(S_n))_{n\in\mathbb{N}}$ point-wise converges almost surely and

$$\mathbb{E}_{s_0}^{\sigma,\tau}[\lim_n val(S_{\min(n,T)})] \geq val(s_0).$$

Proof.
Immediately follows from Proposition 2.22 and Lemma 2.12. ∎

We have shown a few desirable properties of the game under the hypothesis that Max plays with a locally-optimal strategy. Alas, as mentioned before, a locally-optimal strategy may be far from optimal. Nevertheless there exist strategies that are ϵ-optimal and at the same time locally-optimal. This is the following lemma.

Lemma 2.27.
Let $\mu > 0$ and $a \in \mathbf{A}(s)$, a value-changing action in state $s \in \mathbf{S_1}$. There exists a μ-optimal strategy for Max that never plays action a.

Proof.
Let $\mathcal{G}'$ be a game that is identical to $\mathcal{G}$ except that $a \notin \mathbf{A}(s)$ — action a is not enabled in state s in the game $\mathcal{G}'$.[5] Since it is an action of Max that is missing in $\mathcal{G}'$ we have

$$\forall t \in \mathbf{S}, val(\mathcal{G}, t) \geq val(\mathcal{G}', t).$$

We will prove the inverse inequality, *i.e.*

$$\forall t \in \mathbf{S}, val(\mathcal{G}', t) \geq val(\mathcal{G}, t). \tag{2}$$

We split this in two cases, for $t = s$ and for $t \neq s$.

- Case 1. We prove $val(\mathcal{G}', s) \geq val(\mathcal{G}, s)$.

  Let

  $$d = val(\mathcal{G}, s) - \sum_{t\in\mathbf{S}} p(s,a)(t) \cdot val(\mathcal{G}, t) > 0,$$

5 if in game $\mathcal{G}$, $\mathbf{A}(s)$ is a singleton action a, then it cannot be value-changing, therefore in the game $\mathcal{G}'$, $\mathbf{A}(s)$ is nonempty after removing a

$\mu > 0$, and τ a strategy for Min that follows a strategy τ' which is μ-optimal in $\mathcal{G}'$ as long as Max does not play action a in s, and if he does, τ switches definitely to another strategy τ'' that is $\frac{d}{2}$-optimal in $\mathcal{G}$. Define

$$\mathcal{E}_{opt} = \{\forall n, S_n = s \implies A_n \neq a\},$$

the event that a is never played. We will prove that for all σ and $t \in \mathbf{S}$,

$$\mathbb{E}_t^{\sigma,\tau}[f \mid \mathcal{E}_{opt}] \leq val(\mathcal{G}', t) + \mu \tag{3}$$

$$\mathbb{E}_t^{\sigma,\tau}[f \mid \neg \mathcal{E}opt] \leq val(\mathcal{G}, s) - d + \frac{d}{2}. \tag{4}$$

For (3), observe that in infinite histories of $\mathcal{E}_{opt}$ the game is played solely in $\mathcal{G}'$, therefore, even though σ is a strategy in $\mathcal{G}$ it has the same behavior as the strategy σ' in the game $\mathcal{G}'$ defined as

$$\sigma'(h)(b) = \mathbb{P}_{s_0}^{\sigma,\tau}[hb \mid h \cap \mathcal{E}_{opt}],$$

for all $h = s_0 \cdots s_n \in \mathbf{S}(\mathbf{AS})^*$, and $b \in \mathbf{A}(s_n)$. From here, since τ never has to switch to τ'' we have

$$\mathbb{E}_t^{\sigma,\tau}[f \mid \mathcal{E}_{opt}] = \mathbb{E}_t^{\sigma',\tau'}[f],$$

where the first expectation is in the game $\mathcal{G}$ and the second is in the game $\mathcal{G}'$. Now (3) follows from τ' being μ-optimal.

When Max plays the action a, τ switches to the strategy that is $\frac{d}{2}$-optimal in $\mathcal{G}$, and the value decreases by d because of the choice of the value-changing action a, hence (4). With (3) and (4) we have shown that for all strategies σ for Max, $t \in \mathbf{S}$ and $\mu > 0$,

$$\mathbb{E}_t^{\sigma,\tau}[f] \leq \max(val(\mathcal{G}', t) + \mu, val(\mathcal{G}, s) - \frac{d}{2}).$$

Taking $t = s$ and the supremum over strategies σ, from the inequality above we conclude $val(\mathcal{G}', s) \geq val(\mathcal{G}, s)$.

- Case 2. We prove that for all $t \in \mathbf{S}$ different from s, $val(\mathcal{G}', t) \geq val(\mathcal{G}, t)$. Let

  $$\mathcal{E}_\sigma = \{\exists n, S_n = s \wedge \sigma(S_0 \cdots S_n)(a) > 0\},$$

  the event that at some point the state s is reached and the strategy σ wants to play action a with some nonzero probability. Let $\mu > 0$, and for all strategies σ define σ_s to be the strategy in $\mathcal{G}'$ that plays according to σ as long as the latter is not about to play action a with some nonzero probability, in which case it switches to the strategy σ' that is μ-optimal in $\mathcal{G}'$.

Define τ to be the strategy that follows some strategy τ' that is μ-optimal in $\mathcal{G}'$ as long as Max does not play the action a from s, in which case τ switches to a strategy τ'' that is μ-optimal in $\mathcal{G}$.

For all σ, since the pairs of strategies σ, τ play the same as σ_s, τ' respectively up to the date n in the event $\mathcal{E}_\sigma$, we can write

$$c_\sigma = \mathbb{P}_t^{\sigma,\tau}[\mathcal{E}_\sigma] = \mathbb{P}_t^{\sigma_s,\tau'}[\mathcal{E}_\sigma].$$

From the definition of τ, σ_s and Case 1, for all strategies σ we have,

$$\begin{aligned}\mathbb{E}_t^{\sigma,\tau}[f \mid \mathcal{E}_\sigma] &\leq val(\mathcal{G}, s) + \mu = val(\mathcal{G}', s) + \mu \\ \mathbb{E}_t^{\sigma_s,\tau}[f \mid \mathcal{E}_\sigma] &\geq val(\mathcal{G}', s) - \mu.\end{aligned}$$

Whose combination yields

$$\mathbb{E}_t^{\sigma,\tau}[f \mid \mathcal{E}_\sigma] \leq \mathbb{E}_t^{\sigma_s,\tau}[f \mid \mathcal{E}_\sigma] + 2\mu. \tag{5}$$

To conclude we decompose $\mathbb{E}_t^{\sigma,\tau}[f]$ on the event $\mathcal{E}_\sigma$ and make the following observation, for all strategies σ,

$$\begin{aligned}&c_\sigma \mathbb{E}_t^{\sigma,\tau}[f \mid \mathcal{E}_\sigma] + (1 - c_\sigma)\mathbb{E}_t^{\sigma,\tau}[f \mid \neg\mathcal{E}_\sigma] && (6)\\ &\leq c_\sigma(\mathbb{E}_t^{\sigma_s,\tau}[f \mid \mathcal{E}_\sigma] + 2\mu) + (1 - c_\sigma)\mathbb{E}_t^{\sigma_s,\tau}[f \mid \neg\mathcal{E}_\sigma] && (7)\\ &= \mathbb{E}_t^{\sigma_s,\tau}[f] + 2\mu c_\sigma && (8)\\ &= \mathbb{E}_t^{\sigma_s,\tau'}[f] + 2\mu c_\sigma \leq val(\mathcal{G}', t) + \mu(2c_\sigma + 1). && (9)\end{aligned}$$

Here (7) comes from (5) and because σ and σ_s coincide in the infinite histories of $\neg\mathcal{E}_\sigma$, and (9) because σ_s never plays the action a, hence τ plays according to τ' that is μ-optimal in the game $\mathcal{G}'$. Taking the supremum over all σ finishes this case.

We have proved that for all $t \in \mathbf{S}$, $val(\mathcal{G}, t) = val(\mathcal{G}', t)$, so there is a μ-optimal strategy that never plays action a. ∎

From this lemma, by induction on the value-changing actions follows this corollary.

Corollary 2.28.
For all $\mu > 0$, both players have μ-optimal strategies that are locally-optimal.

2.4.3 *Finitely many resets*

The desirable properties that we have gathered in the previous section will now be used to demonstrate that playing with the reset strategy that is based on a strategy that is both ϵ-optimal and locally-optimal ensures that only finitely many weaknesses occur, almost surely. The reset strategy resets the memory only when a weakness occurs, therefore only finitely many weaknesses occur, almost surely.

Proving this will require a couple of steps.

- First, we will show that for σ that is ϵ-optimal and τ locally-optimal, there is nonzero probability (for small enough ϵ) that no σ-weakness occurs.
- Second, when playing with the reset strategy $\hat{\sigma}$ based on some ϵ-optimal strategy σ that is locally-optimal, against all τ, it is ensured that there exists n such that the probability that there are two σ-weakness after date[6] n is bounded away from 1. The intermediate step is to demonstrate the same statement for strategies τ_n. These are strategies that play according to τ up to a date n, after which they only play value-preserving actions.

For the sake of clarity we will use English to refer to certain events. Let us formally state their definitions in terms of Definition 2.19. The event *there is no σ-weakness after date n* is defined as

$$\{\forall m > n, \neg \mathcal{W}_\sigma(S_0 \cdots S_m)\}.$$

The event *there are two σ-weaknesses after date n* is defined as

$$\{\exists m, m', m > m' > n, \mathcal{W}_\sigma(S_0 \cdots S_m) \wedge \mathcal{W}_\sigma(S_0 \cdots S_{m'})\}.$$

For $n \in \mathbb{N}$ and $s_0 \cdots s_m \in \mathbf{S}(\mathbf{AS})^*$, define the boolean-valued function **wa**[7]

$$\mathbf{wa}(n, s_0 \cdots s_m) = (m > n) \wedge (\mathcal{W}_\sigma(s_0 \cdots s_m)) \wedge (\delta_\sigma(s_0 \cdots s_{m-1}) \leq n),$$

that characterizes finite histories that are the first weakness after the date n.

Let us start with the first step.

Lemma 2.29.
Let σ be an ϵ-optimal strategy for Max. There exists $\mu(\epsilon) > 0$ such that for all $s_0 \in \mathbf{S}$ and locally-optimal strategies τ for Min,

$$\mathbb{P}_{s_0}^{\sigma,\tau}[\{\exists n, S_0 \cdots S_n \text{ is a } \sigma\text{-weakness}\}] \leq 1 - \mu(\epsilon).$$

6 the *date* is the number of turns
7 read *weakness after*

We will prove this lemma by contradiction. For a small enough ϵ, if we assume that almost surely there will be a σ-weakness, although σ is ϵ-optimal, we can construct a strategy for Min that takes advantage of this weakness resulting in a less then ϵ-optimal performance from σ.

Proof.
Define the random variable F_σ (the date of the first σ-weakness) as

$$F_\sigma = \min\{n \in \mathbb{N} \mid \mathcal{W}_\sigma(S_0 \cdots S_n)\},$$

with the convention that $\min \emptyset = \infty$. Let τ be a strategy for Min, $s_0 \in \mathbf{S}$, M and m the upper and lower bound of the payoff function f respectively. Define τ' to be the strategy that plays identically to τ as long as no weakness occurs, and if it does it switches definitely to a $\frac{\epsilon}{2}$-response τ''.

Let us explain what we mean by $\frac{\epsilon}{2}$-response. Suppose that $\mathcal{W}_\sigma(s_0 \cdots s_n)$ and that no strict prefix of $s_0 \cdots s_n$ is a σ-weakness. By definition of the σ-weakness, we have

$$\inf_\tau \mathbb{E}_{s_n}^{\sigma[s_0 \cdots s_n],\tau}\big[f[s_0 \cdots s_n]\big] < val(s_n) - 2\epsilon,$$

and τ'' is chosen such that

$$\mathbb{E}_{s_n}^{\sigma[s_0 \cdots s_n],\tau''}\big[f[s_0 \cdots s_n]\big] \leq val(s_n) - 2\epsilon + \frac{\epsilon}{2}. \tag{10}$$

Since τ and τ' coincide up to the date of the first weakness, we write

$$c_\sigma = \mathbb{P}_{s_0}^{\sigma,\tau}[\{F_\sigma = \infty\}] = \mathbb{P}_{s_0}^{\sigma,\tau'}[\{F_\sigma = \infty\}].$$

Decomposing on the events $\{F_\sigma = \infty\}$ and $\{F_\sigma < \infty\}$, we have

$$\begin{aligned}\mathbb{E}_{s_0}^{\sigma,\tau'}[f] &= (1 - c_\sigma)\mathbb{E}_{s_0}^{\sigma,\tau'}[f \mid F_\sigma < \infty] + c_\sigma \mathbb{E}_{s_0}^{\sigma,\tau'}[f \mid F_\sigma = \infty] \\ &\leq (1 - c_\sigma)\mathbb{E}_{s_0}^{\sigma,\tau'}[f \mid F_\sigma < \infty] + c_\sigma M.\end{aligned}$$

Combining the above with the fact that σ is ϵ-optimal we have

$$val(s) - \epsilon \leq (1 - c_\sigma)\mathbb{E}_{s_0}^{\sigma,\tau'}[f \mid F_\sigma < \infty] + c_\sigma M. \tag{11}$$

From the definition of the strategies τ' and τ'', and the assumption that f is shift-invariant it follows that

$$\begin{aligned}&\mathbb{E}_{s_0}^{\sigma,\tau'}[f \mid F_\sigma < \infty] \\ &= \sum_{\mathbf{wa}(0,s_0 \cdots s_n)} \mathbb{P}_{s_0}^{\sigma,\tau'}[s_0 \cdots s_n \mid F_\sigma < \infty]\mathbb{E}_{s_n}^{\sigma[s_0 \cdots s_n],\tau''}[f] \\ &\leq \sum_{\mathbf{wa}(0,s_0 \cdots s_n)} \mathbb{P}_{s_0}^{\sigma,\tau'}[s_0 \cdots s_n \mid F_\sigma < \infty](val(s_n) - 2\epsilon + \frac{\epsilon}{2}) \\ &= \mathbb{E}_{s_0}^{\sigma,\tau'}[val(S_{F_\sigma}) \mid F_\sigma < \infty] - \frac{3}{2}\epsilon.\end{aligned}$$

The sums above are over finite histories $s_0 \cdots s_n \in \mathbf{S}(\mathbf{AS})^*$, for which $\mathbf{wa}(s_0 \cdots s_n)$ holds. Plugging the above into (11) we have

$$
\begin{aligned}
& val(s) - \epsilon \\
& \leq (1 - c_\sigma)\mathbb{E}_{s_0}^{\sigma,\tau'}[val(S_{F_\sigma}) \mid F_\sigma < \infty] - \frac{3}{2}\epsilon(1 - c_\sigma) + c_\sigma M && (12)\\
& = \mathbb{E}_{s_0}^{\sigma,\tau'}[val(S_{F_\sigma})] + c_\sigma(M - \mathbb{E}_{s_0}^{\sigma,\tau'}[val(S_{F_\sigma}) \mid F_\sigma = \infty] && (13)\\
& \qquad\qquad - \frac{3}{2}\epsilon(1 - c_\sigma)) \\
& \leq val(s_0) + c_\sigma(M - \mathbb{E}_{s_0}^{\sigma,\tau'}[val(S_{F_\sigma}) \mid F_\sigma = \infty]) - \frac{3}{2}\epsilon(1 - c_\sigma) && (14)\\
& \leq val(s_0) + c_\sigma(M - m) - \frac{3}{2}\epsilon(1 - c_\sigma). && (15)
\end{aligned}
$$

We have (13) from the decomposition of $\mathbb{E}_{s_0}^{\sigma,\tau'}[val(S_{F_\sigma})]$ on the event $\{F_\sigma < \infty\}$. While (14) comes from the following. The random variable F_σ is a stopping time with respect to the filtration that is induced by the process $(S_n)_{n\in\mathbb{N}}$, and since the strategy τ' plays only value-preserving actions up to the first weakness the process $(val(S_n))_{n\in\mathbb{N}}$ is a submartingale. Applying Lemma 2.12 we have $\mathbb{E}_{s_0}^{\sigma,\tau'}[val(S_{F_\sigma})] \leq val(s_0)$.

Now a uniform bound that does not depend on the choice of the strategy τ follows

$$\mu(\epsilon) = \frac{\epsilon}{2(M - m + \frac{3}{2}\epsilon)} \leq c_\sigma.$$

■

Now we will prove something similar but for the reset strategy $\hat{\sigma}$ that is based on some ϵ-optimal and locally-optimal strategy σ. In fact we will prove that there exists some $n \in \mathbb{N}$ such that there is some nonzero probability that there do not exist two σ-weaknesses after date n. Even though we are playing with the reset strategy, we still care about σ-weaknesses (in place of say $\hat{\sigma}$-weaknesses) because they tell us when the reset strategy resets its memory.

We will prove the above, first for a class of strategies of Min that after some date are sure to play value-preserving actions, then lift this to all strategies of Min, using the fact that if one of the players plays with a locally-optimal strategy then he forces the other one to do so as well.

First let us insert a small proposition. By $\mathbb{1}_{\mathcal{E}}$, denote the indicator function for the event $\mathcal{E}$.

Proposition 2.30.
Let $\mathcal{E}$ be an event, and σ_1, σ_2 two strategies for Max, such that for all prefixes p of an infinite history in $\mathcal{E}$, $\sigma_1(p) = \sigma_2(p)$. For all payoff functions f, strategies τ and $s_0 \in \mathbf{S}$

$$\mathbb{E}_{s_0}^{\sigma_1,\tau}[f \cdot \mathbb{1}_{\mathcal{E}}] = \mathbb{E}_{s_0}^{\sigma_2,\tau}[f \cdot \mathbb{1}_{\mathcal{E}}].$$

Proof.
It is immediate if f is an indicator function itself. The class of functions for which the above holds, is closed under linear combinations and limits, therefore we can use the *monotone class theorem* to finish the proof. For a proof of the monotone class theorem see *e.g.* [Durrett, 2010]. ∎

For a state $s \in \mathbf{S}$ denote by

$$\mathbf{vp}(s) \subseteq \mathbf{A}(s),$$

the set of value-preserving actions in s. For a finite set X denote by

$$\mathcal{U}(X),$$

the *uniform* distribution on X.

Let τ be a strategy for Min. For all $n \in \mathbb{N}$ we define τ_n to be the strategy that after the date n plays only value-preserving actions, *i.e.* for all $s_0 \cdots s_m \in \mathbf{S}(\mathbf{AS})^*$, $s_m \in \mathbf{S_2}$,

$$\tau_n(s_0 \cdots s_m) = \begin{cases} \tau(s_0 \cdots s_m) & \text{if } m < n \text{ or} \\ & \forall a, \tau(s_0 \cdots s_m)(a) > 0 \implies (a \in \mathbf{vp}(s_m)) \\ \mathcal{U}(\mathbf{vp}(s_m)) & \text{otherwise.} \end{cases}$$

In the strategy τ_n, the strategy τ is followed in the first n turns. After the first n turns, if τ plays a value-preserving action, then so does τ_n, otherwise it plays some random value preserving action.

Lemma 2.31.
Let σ and τ be two strategies, with σ being ϵ-optimal and locally-optimal. There exists $\mu(\epsilon) > 0$ such that for all $n \in \mathbb{N}$ and $s_0 \in \mathbf{S}$,

$$\mathbb{P}_{s_0}^{\hat{\sigma},\tau_n}[\{\text{there are two } \sigma\text{-weaknesses after the date } n\}] \leq 1 - \mu(\epsilon).$$

Make note that the statement of the lemma above is about the reset strategy $\hat{\sigma}$, instead of σ.

Proof.
We will define two random variables. For $n \in \mathbb{N}$, let

$$F_n = \min\{m > n \mid \mathcal{W}(S_0 \cdots S_m)\},$$

with the convention $\min \emptyset = \infty$. This is the *first* weakness that occurs strictly after the date n. Define

$$F_n^2 = F_{F_n},$$

the second weakness that occurs strictly after the date n, with the convention $F_\infty = \infty$.

We first show that there exists $\mu(\epsilon) > 0$ such that for all $n \in \mathbb{N}$ and $s_0 \in \mathbf{S}$,

$$\mathbb{P}_{s_0}^{\hat{\sigma},\tau_n}\left[\{F_n^2 < \infty\} \mid F_n < \infty\right] \leq 1 - \mu(\epsilon). \tag{16}$$

As above, define the boolean-valued function **wa**,

$$\mathbf{wa}(n, s_0 \cdots s_m) = (m > n) \wedge (\mathcal{W}_\sigma(s_0 \cdots s_m)) \wedge (\delta_\sigma(s_0 \cdots s_{m-1}) \leq n),$$

for $n \in \mathbb{N}$ and $s_0 \cdots s_m \in \mathbf{S}(\mathbf{AS})^*$. Fix $\mu(\epsilon)$ from Lemma 2.29. Then (16) is a consequence of the following

$$\begin{aligned}
&\mathbb{P}_{s_0}^{\hat{\sigma},\tau_n}[\{F_n^2 < \infty\} \mid F_n < \infty] \\
&= \sum_{\substack{h=s_0\cdots s_m \\ \mathbf{wa}(n,h)}} \mathbb{P}_{s_0}^{\hat{\sigma},\tau_n}[\{F_n^2 < 2\} \mid h \wedge F_n < \infty] \cdot \mathbb{P}_{s_0}^{\hat{\sigma},\tau_n}[h \mid F_n < \infty] && (17)\\
&= \sum_{\substack{h=s_0\cdots s_m \\ \mathbf{wa}(n,h)}} \mathbb{P}_{s_0}^{\hat{\sigma},\tau_n}[\{F_n^2 < \infty\} \mid h] \cdot \mathbb{P}_{s_0}^{\hat{\sigma},\tau_n}[h \mid F_n < \infty] && (18)\\
&= \sum_{\substack{h=s_0\cdots s_m \\ \mathbf{wa}(n,h)}} \mathbb{P}_{s_m}^{\hat{\sigma},\tau_n[h]}[\{F_0 < \infty\}] \cdot \mathbb{P}_{s_0}^{\hat{\sigma},\tau_n}[h \mid F_n < \infty] && (19)\\
&= \sum_{\substack{h=s_0\cdots s_m \\ \mathbf{wa}(n,h)}} \mathbb{P}_{s_m}^{\sigma,\tau_n[h]}[\{F_0 < \infty\}] \cdot \mathbb{P}_{s_0}^{\hat{\sigma},\tau_n}[h \mid F_n < \infty] && (20)\\
&\leq 1 - \mu(\epsilon). && (21)
\end{aligned}$$

We have (17) and (18) because

$$\{F_n < \infty\} = \bigcup_{\substack{h=s_0\cdots s_m \\ \mathbf{wa}(n,h)}} h(\mathbf{AS})^\omega.$$

We have (19) from the definition of the reset strategy, if $\mathbf{wa}(n,h)$ then $\hat{\sigma}[h] = \hat{\sigma}$. We have Equation (20) from (2.30) because σ and $\hat{\sigma}$ coincide up to the first σ-weakness. Finally (21) is from the definition of τ_n, since $|h| \geq n$, $\tau_n[h]$ is locally-optimal so we can use Lemma 2.29.

To finish the proof, we have

$$\mathbb{P}_{s_0}^{\hat{\sigma},\tau_n}[F_n^2 < \infty] = \mathbb{P}_{s_0}^{\hat{\sigma},\tau_n}[F_n^2 \mid F_n < \infty] \cdot \mathbb{P}_{s_0}^{\hat{\sigma},\tau_n}[F_n < \infty] \leq 1 - \mu(\epsilon). \qquad \blacksquare$$

Now we are able to strengthen this result so that it holds for any strategy chosen by Min. We have to do this strengthening because there is no guarantee that Min always plays intelligently, he might in general play actions which do not preserve the value.

Lemma 2.32.
Let σ be a strategy for Max that is ϵ-optimal and locally-optimal. For all τ and $s_0 \in \mathbf{S}$ there exists $n \in \mathbb{N}$ such that

$$\mathbb{P}_{s_0}^{\hat{\sigma},\tau}[\{\text{there are two } \sigma\text{-weaknesses after date } n\}] < 1.$$

Proof.
Let L be the random variable taking values in $\mathbb{N} \cup \{\infty\}$ that maps to the date of the last value-changing action that has been played, if it exists, and it maps to ∞ if it does not. The strategies τ and τ_n coincide on all infinite histories where the last value-changing action is played before the date n, *i.e.* the event $\{L < n\}$. Hence for all $n \in \mathbb{N}$ and event $\mathcal{E}$

$$\begin{aligned}\mathbb{P}_{s_0}^{\hat{\sigma},\tau}[\mathcal{E}] = \mathbb{P}_{s_0}^{\hat{\sigma},\tau}[\{L < n\}] \cdot \mathbb{P}_{s_0}^{\hat{\sigma},\tau_n}[\mathcal{E} \mid L < n] \\ + \mathbb{P}_{s_0}^{\hat{\sigma},\tau}[\{L \geq n\}] \cdot \mathbb{P}_{s_0}^{\hat{\sigma},\tau}[\mathcal{E} \mid L \geq n].\end{aligned}$$

The strategy $\hat{\sigma}$ is locally-optimal, so from Corollary 2.25 we have

$$\lim_n \mathbb{P}_{s_0}^{\hat{\sigma},\tau}[\{L < n\}] = 1,$$

and consequently

$$\lim_n \mathbb{P}_{s_0}^{\hat{\sigma},\tau_n}[\mathcal{E}] = \mathbb{P}_{s_0}^{\hat{\sigma},\tau}[\mathcal{E}].$$

Fix $\mu(\epsilon)$ from Lemma 2.31 and $n \in \mathbb{N}$ such that for all $\mathcal{E}$

$$|\mathbb{P}_{s_0}^{\hat{\sigma},\tau}[\mathcal{E}] - \mathbb{P}_{s_0}^{\hat{\sigma},\tau_n}[\mathcal{E}]| < \mu(\epsilon).$$

Now take $n' > n$ and let $\mathcal{E} = \{\text{there are two } \sigma\text{-weaknesses after date } n'\}$ and apply Lemma 2.31. $\blacksquare$

Finally we can prove that almost surely, only finitely many times the reset strategy will reset its memory.

Lemma 2.33 (Finitely many resets).
Let σ be a strategy that is locally-optimal and ϵ-optimal. For all strategies τ and $s_0 \in \mathbf{S}$,

$$\mathbb{P}_{s_0}^{\hat{\sigma},\tau}[\{\exists n, \text{there is no } \sigma\text{-weakness after date n}\}] = 1$$

Proof.
Let

$$L = \lim_n \delta_\sigma(S_0 \cdots S_n) \in \mathbb{N} \cup \{\infty\}.$$

The random variable L is well-defined since the $\delta_\sigma(S_0 \cdots S_n)$ is point-wise increasing as a function of n. In other words L is the date of the last σ-weakness, if it exists, otherwise it is equal to ∞.

Let $\epsilon' > 0$ and choose τ and s_0 such that

$$\sup_{\tau', s'} \mathbb{P}_{s'}^{\hat{\sigma},\tau'}[\{L = \infty\}] \leq \mathbb{P}_{s_0}^{\hat{\sigma},\tau}[\{L = \infty\}] + \epsilon'. \tag{22}$$

Fix $n \in \mathbb{N}$ to be such that according to Lemma 2.32 we have

$$\mu = \mathbb{P}_{s_0}^{\hat{\sigma},\tau}[\{F_n < \infty\}] < 1,$$

where F_n is the random variable that gives *the date of the first weakness strictly after n, i.e.*

$$F_n = \min\{m > n \mid \mathcal{W}(S_0 \cdots S_m)\}.$$

Note that we can apply Lemma 2.32 for the event $\{F_n < \infty\}$, because if the event of having at most one weakness after date n is nonzero, we can choose some $n' > n$ such that the event of having one weakness after n' is nonzero.

We now have

$$\begin{aligned}
&\mathbb{P}_s^{\hat{\sigma},\tau}[\{L = \infty\}] \\
&= \mathbb{E}_s^{\hat{\sigma},\tau}\left[\mathbb{P}_s^{\hat{\sigma},\tau}[\{L = \infty\} \mid F_n, S_0, \ldots, S_{F_n}]\right] && (23)\\
&= \mathbb{E}_s^{\hat{\sigma},\tau}\left[\mathbb{1}_{F_n<\infty} \cdot \mathbb{P}_s^{\hat{\sigma},\tau}[\{L = \infty\} \mid F_n, S_0, \ldots, S_{F_n}]\right] && (24)\\
&= \mathbb{E}_s^{\hat{\sigma},\tau}\left[\mathbb{1}_{F_n<\infty} \cdot \mathbb{P}_{S_{F_n}}^{\hat{\sigma},\tau[S_0\cdots S_{F_n}]}[\{L = \infty\}]\right] && (25)\\
&\leq \mathbb{E}_s^{\hat{\sigma},\tau}\left[\mathbb{1}_{F_n<\infty} \cdot (\mathbb{P}_s^{\hat{\sigma},\tau}[\{L = \infty\}] + \epsilon')\right] && (26)\\
&= \mu\left(\mathbb{P}_s^{\hat{\sigma},\tau}[\{L = \infty\}] + \epsilon'\right) && (27)
\end{aligned}$$

We have (23) from a basic property of conditional expectations. Then (24) follows from the fact that $\mathbb{P}_s^{\hat{\sigma},\tau}[\{F_n < \infty\} \mid L = \infty] = 1$. From the definition of

the reset strategy comes (25), and (26) follows from (22). Consequently, since $\mu < 1$ we get

$$\mathbb{P}_s^{\hat{\sigma},\tau}[\{L = \infty\}] \leq \frac{\mu}{1-\mu}\epsilon'.$$

Now for all s_0 and τ''

$$\begin{aligned}\mathbb{P}_{s_0}^{\hat{\sigma},\tau''} &\leq \sup_{\tau',s'} \mathbb{P}_{s'}^{\hat{\sigma},\tau'}[\{L = \infty\}] \\ &\leq \mathbb{P}_s^{\hat{\sigma},\tau}[\{L = \infty\}] + \epsilon' \\ &\leq \frac{\epsilon'}{1-\mu}.\end{aligned}$$

This holds for all $\epsilon' > 0$, therefore $\mathbb{P}_{s_0}^{\hat{\sigma},\tau''}[\{L = \infty\}] = 0$. ∎

2.4.4 *The reset strategy is 2ϵ-subgame-perfect*

When playing with the reset strategy, we see that on almost all possible infinite histories, there is some date n after which there will be no more resets of the memory, since there will be no more weaknesses, and consequently the strategy will be 2ϵ-subgame-perfect.

In order to show that the reset strategy is 2ϵ-subgame-perfect for all finite histories, we will first prove that it is ϵ-optimal. This will be done in two steps, where the first step is proving it for strategies that reset the memory only up to some date n.

Let σ be a strategy for Max, we define the strategy based on σ that resets the memory only up to date n, denoted $\hat{\sigma}_n$ as

$$\hat{\sigma}_n(s_0 \cdots s_m) = \sigma(s_{\delta_\sigma(s_0 \cdots s_{m\wedge n})} \cdots s_m),$$

where $m \wedge n = \min(m, n)$.

Lemma 2.34.
Let σ be a strategy for Max that is ϵ-optimal and locally-optimal. For all $n \in \mathbb{N}$, $\hat{\sigma}_n$ is ϵ-optimal.

Proof.
The proof is by induction on n.

Base case. By definition since $\hat{\sigma}_0 = \sigma$.

Induction step. Let τ be a strategy for Min and $s_0 \in \mathbf{S}$. Define τ' to be the strategy that follows τ except if there is a weakness at date $n+1$, *i.e.* the

event $\{\mathcal{W}_\sigma(S_0 \cdots S_{n+1})\}$ in which case it resets definitely to an $\frac{\epsilon}{2}$-response τ''. Define the random variable

$$\mathcal{L}_n = \mathcal{W}_\sigma(S_0 \cdots S_n).$$

We will decompose the expected values on the event of a weakness at date $n+1$. Denote by $\mathbf{W}$ the set of all finite histories of $n+1$ turns, where a weakness occurs,

$$\mathbf{W} = \{s_0 \cdots s_{n+1} \in \mathbf{S}(\mathbf{AS})^* \mid \mathcal{W}_\sigma(s_0 \cdots s_{n+1})\}.$$

Hence

$$\{\mathcal{L}_{n+1}\} = \bigcup_{h \in \mathbf{W}} h(\mathbf{AS})^\omega.$$

From here,

$$\begin{aligned}
&\mathbb{E}_{s_0}^{\hat{\sigma}_{n+1},\tau}[f] \\
&= \mathbb{E}_{s_0}^{\hat{\sigma}_{n+1},\tau}[\mathbb{1}_{\mathcal{L}_{n+1}} \cdot f] + \mathbb{E}_{s_0}^{\hat{\sigma}_{n+1},\tau}[\mathbb{1}_{\neg\mathcal{L}_{n+1}} \cdot f] \\
&= \sum_{h=s_0\cdots s_{n+1} \in \mathbf{W}} \mathbb{P}_{s_0}^{\hat{\sigma}_{n+1},\tau}[h] \cdot \mathbb{E}_{s_{n+1}}^{\sigma,\tau[h]}[f] + \mathbb{E}_{s_0}^{\hat{\sigma}_{n+1},\tau}[\mathbb{1}_{\neg\mathcal{L}_{n+1}} \cdot f] \\
&\geq \sum_{h=s_0\cdots s_{n+1} \in \mathbf{W}} \mathbb{P}_{s_0}^{\hat{\sigma}_{n+1},\tau}[h] \cdot (val(s_{n+1}) - \epsilon) + \mathbb{E}_{s_0}^{\hat{\sigma}_{n+1},\tau}[\mathbb{1}_{\neg\mathcal{L}_{n+1}} \cdot f],
\end{aligned}$$

from the definition of $\hat{\sigma}_{n+1}$, shift-invariance of f and the fact that σ is ϵ-optimal.

In the other hand

$$\begin{aligned}
&\mathbb{E}_{s_0}^{\hat{\sigma}_n,\tau'}[f] \\
&= \mathbb{E}_{s_0}^{\hat{\sigma}_n,\tau'}[\mathbb{1}_{\mathcal{L}_{n+1}} \cdot f] + \mathbb{E}_{s_0}^{\hat{\sigma}_n,\tau'}[\mathbb{1}_{\neg\mathcal{L}_{n+1}} \cdot f] \\
&= \sum_{h=s_0\cdots s_{n+1} \in \mathbf{W}} \mathbb{P}_{s_0}^{\hat{\sigma}_n,\tau'}[h] \cdot \mathbb{E}_{s_{n+1}}^{\hat{\sigma}_n[s_0\cdots s_{n+1}],\tau''}[f] + \mathbb{E}_{s_0}^{\hat{\sigma}_n,\tau'}[\mathbb{1}_{\neg\mathcal{L}_{n+1}} \cdot f] \\
&\leq \sum_{h=s_0\cdots s_{n+1} \in \mathbf{W}} \mathbb{P}_{s_0}^{\hat{\sigma}_n,\tau'}[h]\big(val(s_{n+1}) - 2\epsilon + \frac{\epsilon}{2}\big) + \mathbb{E}_{s_0}^{\hat{\sigma}_n,\tau'}[\mathbb{1}_{\neg\mathcal{L}_{n+1}} \cdot f]
\end{aligned}$$

by construction of τ' and shift-invariance of f.

We can combine the two inequalities above because both pairs of strategies $\hat{\sigma}_n$,$\hat{\sigma}_{n+1}$ and τ,τ' coincide on all infinite histories where no σ-weakness occurs at date $n+1$ and on every action before the date $n+1$. Therefore the second terms of the inequalities above are equal hence

$$\mathbb{E}_{s_0}^{\hat{\sigma}_n,\tau'}[f] \leq \mathbb{E}_{s_0}^{\hat{\sigma}_{n+1},\tau}[f].$$

From the induction hypothesis $\hat{\sigma}$ is ϵ-optimal thus

$$\mathbb{E}_{s_0}^{\hat{\sigma}_{n+1},\tau}[f] \geq val(s_0) - \epsilon.$$

■

Being done with this intermediate step we move on to show that the reset strategy is ϵ-optimal.

Lemma 2.35.
Let σ be a strategy for Max that is ϵ-optimal and locally-optimal. Then $\hat{\sigma}$ is ϵ-optimal.

Proof.
Let m and M be respectively the lower and upper bound of f. Define

$$L = \lim_n \delta_\sigma(S_0 \cdots S_n) \in \mathbb{N} \cup \{\infty\},$$

the date of the last σ-weakness if it exists, and otherwise ∞. Let $s_0 \in \mathbf{S}$ and τ a strategy for Min, from Lemma 2.34 we have

$$\begin{aligned} val(s_0) - \epsilon &\leq \mathbb{E}_{s_0}^{\hat{\sigma}_n,\tau}[f \cdot \mathbb{1}_{L \leq n}] + \mathbb{E}_{s_0}^{\hat{\sigma}_n,\tau}[f \cdot \mathbb{1}_{L\, n}] \\ &\leq \mathbb{E}_{s_0}^{\hat{\sigma}_n,\tau}[f \cdot \mathbb{1}_{L \leq n}] + M \cdot \mathbb{P}_{s_0}^{\hat{\sigma}_n,\tau}[\{L > n\}]. \end{aligned}$$

Since σ and $\hat{\sigma}_n$ coincide upon all infinite histories in $\{L \leq n\}$, using Proposition 2.30 we have

$$\begin{aligned} \mathbb{E}_{s_0}^{\hat{\sigma},\tau}[f] - \mathbb{E}_{s_0}^{\hat{\sigma},\tau}[f \cdot \mathbb{1}_{L>n}] &= \mathbb{E}_{s_0}^{\hat{\sigma},\tau}[f \cdot \mathbb{1}_{L \leq n}] \\ &= \mathbb{E}_{s_0}^{\hat{\sigma}_n,\tau}[f \cdot \mathbb{1}_{L>n}] \\ &\geq val(s_0) - \epsilon - M \cdot \mathbb{P}_{s_0}^{\hat{\sigma}_n,\tau}[\{L > n\}]. \end{aligned}$$

Applying Proposition 2.30 for the constant payoff function that maps to 1 we have

$$\mathbb{P}_{s_0}^{\hat{\sigma},\tau}[\{L \leq n\}] = \mathbb{P}_{s_0}^{\hat{\sigma}_n,\tau}[\{L \leq n\}],$$

therefore

$$\mathbb{E}_{s_0}^{\hat{\sigma},\tau}[f] \geq val(s_0) - \epsilon - (M - m) \cdot \mathbb{P}_{s_0}^{\hat{\sigma},\tau}[\{L > n\}].$$

From Lemma 2.33 the right-most term above vanishes as n grows, which concludes the proof. ∎

After gathering all the lemmata above we finish this section by proving that the reset strategy is 2ϵ-subgame-perfect.

Lemma 2.36.
Let σ be a strategy for Max, that is ϵ-optimal and locally optimal. The reset strategy $\hat{\sigma}$ is 2ϵ-subgame-perfect.

Proof.
Let

$$h = s_0 \cdots s_n \in \mathbf{S}(\mathbf{AS})^*,$$

we will show that

$$\inf_{\tau} \mathbb{E}_{s_n}^{\hat{\sigma}[h],\tau}[f] \geq val(s_n) - 2\epsilon. \tag{28}$$

In case $\mathcal{W}_\sigma(h)$ we have

$$\inf_{\tau} \mathbb{E}_{s_n}^{\hat{\sigma}[h],\tau}[f] = \inf_{\tau} \mathbb{E}_{s_n}^{\hat{\sigma},\tau}[f] \geq val(s_n) - \epsilon,$$

from the definition of $\hat{\sigma}$ and Lemma 2.35. Therefore assume that $\neg\mathcal{W}_\sigma(h)$, this implies that

$$\delta_\sigma(s_0 \cdots s_n) = \delta_\sigma(s_0 \cdots s_{n-1}),$$

i.e. the date of the last σ-weakness in the finite play $s_0 \cdots s_n$ is the same as the date in the finite play $s_0 \cdots s_{n-1}$ since the assumption is that h is not a σ-weakness. From the definition of a σ-weakness and that of the reset strategy, the assumption $\neg\mathcal{W}_\sigma(h)$ also implies that

$$\inf_{\tau} \mathbb{E}_{s_n}^{\sigma[h'],\tau}[f] \geq val(s_n) - 2\epsilon, \tag{29}$$

where

$$h' = s_{\delta_\sigma(s_0 \cdots s_{n-1})} \cdots s_n.$$

To prove (28), we will proceed by contradiction, assume that there exists some strategy τ such that

$$\mathbb{E}_{s_n}^{\hat{\sigma}[h],\tau}[f] < val(s_n) - 2\epsilon.$$

Now from τ we will construct another strategy τ' such that

$$\mathbb{E}_{s_n}^{\sigma[h'],\tau'}[f] < val(s_n) - 2\epsilon$$

which contradicts (29).

Define τ' to be the strategy that follows τ as long as no weakness occurs, and when it does, it switches to the ϵ-response strategy τ''. Define the random variable

$$L = \lim_n \delta_\sigma(S_0 \cdots S_n) \in \mathbb{N} \cup \{\infty\},$$

the date of the last σ-weakness, if it exists and otherwise ∞. Denote by F_σ the date of the first σ-weakness, *i.e.*

$$F_\sigma = \min\{n \in \mathbb{N} \mid \mathcal{W}_\sigma(S_0 \cdots S_n)\}.$$

Now we have

$$\begin{aligned} val(s_n) - 2\epsilon &> \mathbb{E}_{s_n}^{\hat{\sigma}[h],\tau}[f \cdot \mathbb{1}_{L=0}] + \mathbb{E}_{s_n}^{\hat{\sigma}[h],\tau}[f \cdot \mathbb{1}_{F_\sigma<\infty}] && (30)\\ &= \mathbb{E}_{s_n}^{\sigma[h'],\tau'}[f \cdot \mathbb{1}_{L=0}] + \mathbb{E}_{s_n}^{\hat{\sigma}[h],\tau}[f \cdot \mathbb{1}_{F_\sigma<\infty}] && (31)\\ &= \mathbb{E}_{s_n}^{\sigma[h'],\tau'}[f] - \mathbb{E}_{s_n}^{\sigma[h'],\tau'}[f \cdot \mathbb{1}_{F_\sigma<\infty}] + \mathbb{E}_{s_n}^{\hat{\sigma}[h],\tau}[f \cdot \mathbb{1}_{F_\sigma<\infty}]. && (32)\end{aligned}$$

We have (30) because $\{L = 0\} = \{F_\sigma = \infty\}$ and the assumption on the strategy τ. We have (31) because the pairs of strategies $\hat{\sigma}[h]$,$\sigma[h']$ and τ, τ'coincide up to the first σ-weakness.

For the last two terms of (31) we have

$$\begin{aligned}\mathbb{E}_{s_n}^{\hat{\sigma}[h],\tau}[f \cdot \mathbb{1}_{F_\sigma<\infty}] &= \sum_{\substack{g=t_0\cdots t_m\\ \mathbf{wa}(g)}} \mathbb{P}_{s_n}^{\hat{\sigma}[h],\tau}[g] \cdot \mathbb{E}_{t_m}^{\hat{\sigma},\tau[g]}[f]\\ &\geq \sum_{\substack{g=t_0\cdots t_m\\ \mathbf{wa}(g)}} \mathbb{P}_{s_n}^{\hat{\sigma}[h],\tau}[g] \cdot (val(t_m) - \epsilon)\end{aligned}$$

by the definition of $\hat{\sigma}$ and Lemma 2.35. For the other term we have

$$\begin{aligned}\mathbb{E}_{s_n}^{\sigma[h'],\tau'}[f \cdot \mathbb{1}_{F_\sigma<\infty}] &= \sum_{\substack{g=t_0\cdots t_m\\ \mathbf{wa}(g)}} \mathbb{P}_{s_n}^{\sigma[h'],\tau'}[g] \cdot \mathbb{E}_{t_m}^{\sigma[h'][g],\tau''}[f]\\ &\leq \sum_{\substack{g=t_0\cdots t_m\\ \mathbf{wa}(g)}} \mathbb{P}_{s_n}^{\sigma[h'],\tau'}[g] \cdot \big(val(t_m) - 2\epsilon + \epsilon\big),\end{aligned}$$

from the definition of τ'.

In the probabilities of the cylinders $g(\mathbf{AS})^\omega$ we can freely interchange the strategies $\hat{\sigma}[h]$ and $\sigma[h']$ as well as τ and τ' since they coincide up to the first weakness. Therefore we have

$$\mathbb{E}_{s_n}^{\hat{\sigma}[h],\tau}[f \cdot \mathbb{1}_{F_\sigma<\infty}] \geq \mathbb{E}_{s_n}^{\sigma[h'],\tau'}[f \cdot \mathbb{1}_{F_\sigma<\infty}],$$

which contradicts (29) when replaced into (32). ∎

2.5 HALF-POSITIONALITY

In this section we will prove the main result of the chapter. If the payoff function is submixing and shift-invariant then the game is half-positional: Max has an optimal strategy that is both stationary and pure.

Theorem 2.37.
Let $\mathcal{G}$ be a game equipped with a payoff function that is shift-invariant and submixing, then $\mathcal{G}$ is half-positional.

Proof Idea.
The proof is by induction on the number of actions. We will choose some state and partition the set of actions that are available into two sets, and define two games by restricting the available actions to the two corresponding sets. Then the goal is to show that the value of the original game does not exceed the maximum of the values of the two subgames. If this is true then we can play only actions from one of the sets and win just as much as playing in the original game. The induction hypothesis will imply that the subgames are half-positional so in the original game there will be an optimal strategy that is stationary and pure.

The value of the original game is shown to be no more than the maximum of the values of the two subgames by decomposing the set of outcomes in the original game into the trajectories that eventually remain only in subgame 1, the trajectories that eventually remain only in subgame 2 and the trajectories that switch between the two at infinitely many stages. For each one of them we will show that it is not possible to do better than the maximum of the values of the subgames. △

Let $\mathcal{G} = (\mathcal{A}, f)$ be a game, such that f is shift-invariant and submixing. The proof of Theorem 2.37 is by induction on

$$\mathcal{N} = \sum_{s \in \mathbf{S_1}} (|\mathbf{A}(s)| - 1).$$

Base case. If $\mathcal{N} = 0$, Max has a single strategy and it is stationary, pure and optimal.

Induction step. Assume that the theorem holds for $\mathcal{N} = n$ we will prove it for $\mathcal{N} = n + 1$. Let $s \in \mathbf{S_1}$ such that $\mathbf{A}(s) \geq 2$. Such a state exists because $\mathcal{N} = n + 1$. Partition $\mathbf{A}(s)$ into two nonempty sets $\mathfrak{A}_1$ and $\mathfrak{A}_2$, and let $\mathcal{G}_1$ and $\mathcal{G}_2$ be the games that are identical to $\mathcal{G}$ except that in game $\mathcal{G}_i$, $\mathbf{A}(s) = \mathfrak{A}_i$.

In order to prove Theorem 2.37 it suffices to show that

$$val(\mathcal{G}, s) \leq \max\big(val(\mathcal{G}_1, s), val(\mathcal{G}_2, s)\big). \tag{33}$$

This is because, from the induction hypothesis the games $\mathcal{G}_i$ admit optimal strategies that are stationary and pure. Moreover a strategy in the game $\mathcal{G}_i$ is a strategy in the game $\mathcal{G}$ as well.

Therefore we will proceed with the proof of (33).

For a finite set X, we used the notation X^* for the set of finite words over the alphabet X, and X^ω for the set of infinite words over the alphabet X. Denote by

$$X^\infty = X^* \cup X^\omega,$$

their union.

We will define two projection mappings on the set of finite or infinite histories that start from the state s in the game $\mathcal{G}$ into histories in the games $\mathcal{G}_i$, by deleting appropriate subwords,

$$\pi_1 : s(\mathbf{AS})^\infty \to s(\mathbf{AS})^\infty$$
$$\pi_2 : s(\mathbf{AS})^\infty \to s(\mathbf{AS})^\infty.$$

We give the precise definition of π_1. That of π_2 is symmetrical.

For finite histories, π_1 deletes the subword that are in the game $\mathcal{G}_2$, *i.e.* the ones where the state s is followed by an action in $\mathfrak{A}_2$. Let $h = sa_0s_1a_1 \cdots a_{n-1}s_n \in s(\mathbf{AS})^*$, $\pi(h)$ is the finite history where the following subwords are deleted

- all simple cycles on s starting with an action in $\mathfrak{A}_2$,
- in case the last occurrence of s in h is followed by an action in $\mathfrak{A}_2$, then that suffix is deleted.

Formally let $i_1 < i_2 < \cdots < i_k = \{1 \leq i \leq n \mid s_i = s\}$, the dates where the finite history reaches state s, and for all $1 \leq j < k$ define the subword

$$h_j = s_{i_j} a_{i_j} \cdots a_{i_{j+1}-1},$$

and

$$h_k = s_{i_k} a_{i_k} \cdots a_{n-1} s_n$$

Then

$$\pi_1(h) = \prod_{\substack{1 \leq j \leq k \\ a_{i_j} \in \mathfrak{A}_1}} h_j,$$

where $\prod$ is word concatenation. For infinite histories $h \in s(\mathbf{AS})^\omega$, π_1 is extended naturally as the limit of the sequence $(\pi_1(h_n))_{n\in\mathbb{N}}$ where h_n is the finite history of the first n rounds in h, *i.e.* the prefix of length $2n+1$.

Observe that $\pi_1(h)$, is not necessarily an infinite history, even when h itself is. If the tail of the history is in the game $\mathcal{G}_2$ then π_1 will map to a finite history.

Let us enumerate some other properties of the mappings π_1, and π_2. For all $h = s(\mathbf{AS})^\omega$,

- if $\pi_1(h)$ is finite,
  then h has a suffix which is an infinite history in $\mathcal{G}_2$ starting from s, (34)
- if $\pi_2(h)$ is finite,
  then h has a suffix which is an infinite history in $\mathcal{G}_1$ starting from s, (35)
- if both $\pi_1(h)$ and $\pi_2(h)$ are infinite,
  then in both of them the state s is reached infinitely often, (36)
- if both $\pi_1(h)$ and $\pi_2(h)$ are infinite,
  then h is a shuffle of $\pi_1(h)$ and $\pi_2(h)$. (37)

The properties above follow immediately from the definitions of π_1 and π_2.

From Lemma 2.17 there exist ϵ-subgame-perfect strategies $\tau_1^\#$ and $\tau_2^\#$ in the games $\mathcal{G}_1$ and $\mathcal{G}_2$ respectively. By combining these two we construct a strategy $\tau^\#$, called the *trigger strategy*. It switches between $\tau_1^\#$ and $\tau_2^\#$ depending on the action that was played in the last visit to state s. For a finite history $h = sa_0s_1a_1 \cdots a_{n-1} \in (\mathbf{SA})^*$ let $l(h)$ be the action that was played in the last visit of s. Let $s_n \in \mathbf{S}$ and define

$$\tau^\#(hs_n) = \begin{cases} \tau_1^\#(\pi_1(h)s_n) & \text{if } l(h) \in \mathfrak{A}_1 \\ \tau_2^\#(\pi_2(h)s_n) & \text{if } l(h) \in \mathfrak{A}_2. \end{cases}$$

The trigger strategy is retaining the finite histories in both subgames, and depending on which subgame Max is playing, $\tau^\#$ replies with actions that follow the ϵ-subgame-perfect strategy in that subgame.

Define the three following random variables,

$$\begin{aligned} \Pi &= S_0A_0S_1A_1 \cdots, \\ \Pi_1 &= \pi_1(S_0A_0S_1A_1 \cdots), \\ \Pi_2 &= \pi_2(S_0A_0S_1A_1 \cdots). \end{aligned}$$

We will show that the trigger strategy is ϵ-optimal in $\mathcal{G}$. In particular we will prove that for all strategies σ for Max,

$$\begin{aligned} &\mathbb{E}_s^{\sigma,\tau^\#}[f \mid \Pi_1 \text{ is finite}] \leq val(\mathcal{G}_2, s) + \epsilon, \\ &\mathbb{E}_s^{\sigma,\tau^\#}[f \mid \Pi_2 \text{ is finite}] \leq val(\mathcal{G}_1, s) + \epsilon, \\ &\mathbb{E}_s^{\sigma,\tau^\#}[f \mid \Pi_1 \text{ and } \Pi_2 \text{ are infinite}] \leq \max\big(val(\mathcal{G}_1, s), val(\mathcal{G}_2, s)\big) + \epsilon. \end{aligned}$$

Observe that this is sufficient for (33), so we will finish the proof of Theorem 2.37 with the demonstration of the above inequalities.

Proposition 2.38.
For all strategies σ for Max,

$$\mathbb{E}_s^{\sigma,\tau^\#}[f \mid \Pi_1 \text{ is finite}] \leq val(\mathcal{G}_2, s) + \epsilon,$$
$$\mathbb{E}_s^{\sigma,\tau^\#}[f \mid \Pi_2 \text{ is finite}] \leq val(\mathcal{G}_1, s) + \epsilon.$$

Proof.
We prove just the first inequality, the proof of the second one is symmetric.

Define $\mathbb{P}_2$ a probability measure on the set of infinite histories in the game $\mathcal{G}_2$ as follows, for all measurable events $\mathcal{E}$,

$$\mathbb{P}_2[\mathcal{E}] = \mathbb{P}_s^{\sigma,\tau^\#}[\Pi_2 \in \mathcal{E} \mid \Pi_1 \text{ is finite}].$$

We will show that under this measure, the expected payoff is smaller then $val(\mathcal{G}_2, s) + \epsilon$. Toward this purpose, we define a strategy σ_2 for Max in the game $\mathcal{G}_2$ as follows, for all finite histories $h = s_0 \cdots s_n$, with $s_n \in \mathbf{S_1}$, and $a \in \mathbf{A}(s_n)$,

$$\sigma_2(h)(a) = \mathbb{P}_s^{\sigma,\tau^\#}[ha \preceq \Pi_2 \mid h \preceq \Pi_2 \text{ and } \Pi_1 \text{ is finite}],$$

where $\preceq$ denotes the prefix relation. We see that

$$\sigma_2(h)(a) = \mathbb{P}_2[ha \mid h].$$

Consequently, for all $h = s_0 \cdots s_n$, and $t \in \mathbf{S}$

$$\mathbb{P}_2[ha \mid h] = \begin{cases} \sigma_2(h)(a) & \text{if } s_n \in \mathbf{S_1}, \\ \tau_2^\#(h)(a) & \text{if } s_n \in \mathbf{S_2}, \end{cases}$$
$$\mathbb{P}_2[hat \mid ha] = p(s_n, a)(t).$$

From this, the fact that $\mathbb{P}_2[s(\mathbf{AS})^\omega] = 1$ and the definition of a probability measure given an arena (Section 2.2.2) we conclude that $\mathbb{P}_2$ coincides with the measure $\mathbb{P}_s^{\sigma_2,\tau_2^\#}$. Since $\tau_2^\#$ is ϵ-optimal in $\mathcal{G}_2$, it follows that the expected payoff under the measure $\mathbb{P}_2$ is smaller than $val(\mathcal{G}_2, s) + \epsilon$. ∎

Proposition 2.39.
For all strategies σ for Max,

$$\mathbb{E}_s^{\sigma,\tau^\#}[f \mid \Pi_1 \text{ and } \Pi_2 \text{ are infinite}] \leq \max\left(val(\mathcal{G}_1, s), val(\mathcal{G}_2, s)\right) + \epsilon.$$

Proof.
We will first prove two claims.

Claim 1.
For all strategies σ' for Max in $\mathcal{G}_1$

$$\mathbb{P}_s^{\sigma',\tau_1^\#}[\{f \le val(\mathcal{G}_1, s) + \epsilon\} \mid s \text{ is reached infinitely often}] = 1.$$

Since f is shift-invariant and $\tau_1^\#$ is ϵ-subgame-perfect we have that for all $n \in \mathbb{N}$,

$$\begin{aligned}\mathbb{E}_s^{\sigma',\tau_1^\#}[f \mid S_0, A_0, \dots, S_n] &= \mathbb{E}_s^{\sigma',\tau_1^\#}[f(S_n A_n S_{n+1} \cdots) \mid S_0, A_0, \dots, S_n] \\ &= \mathbb{E}_{S_n}^{\sigma'[S_0 \cdots S_n], \tau_1^\#[S_0 \cdots S_n]}[f] \\ &\le val(S_n) + \epsilon.\end{aligned}$$

Using Levy's 0-1 law (see *e.g.* [Durrett, 2010])

$$\lim_n \mathbb{E}_s^{\sigma',\tau_1^\#}[f \mid S_0, A_0, \dots, S_n] = f(S_0 A_0 S_1 \cdots) \qquad \text{a.s.},$$

consequently

$$\mathbb{P}_s^{\sigma',\tau_1^\#}[\{f \le \liminf_n val(\mathcal{G}_1, S_n) + \epsilon\}] = 1,$$

which proves Claim 1.

Claim 2.
For all strategies σ for Max in $\mathcal{G}$, there exists σ_1 in $\mathcal{G}_1$ such that for any measurable event $\mathcal{E}$ in $\mathcal{G}_1$,

$$\mathbb{P}_s^{\sigma_1,\tau_1^\#}[\mathcal{E}] \ge \mathbb{P}_s^{\sigma,\tau^\#}[\{\Pi_1 \text{ is infinite, and } \Pi_1 \in \mathcal{E}\}]. \tag{38}$$

We define σ_1 as follows, for all $h = s_0 \cdots s_n$, $s_n \in \mathbf{S_1}$, and $a \in \mathbf{A}(s_n)$,

$$\sigma_1(h)(a) = \mathbb{P}_s^{\sigma,\tau^\#}[ha \preceq \Pi_1 \mid h \preceq \Pi_1],$$

if $\mathbb{P}_s^{\sigma_1,\tau^\#}[h \preceq \Pi_1] > 0$, otherwise $\sigma(h)$ is chose arbitrarily.

Let $\mathfrak{E}$ be the set of events for which the claim holds. We will show that the cylinders are in $\mathfrak{E}$ and that it is closed under countable monotone unions and intersections. That the claim holds for all measurable events, then follows from the monotone class theorem.

Indeed $\mathfrak{E}$ contains all the cylinders $h_1(\mathbf{AS})^\omega$ since

$$\mathbb{P}_s^{\sigma_1,\tau_1^\#}[h_1] \ge \mathbb{P}_s^{\sigma,\tau^\#}[h_1 \preceq \Pi_1] \ge \mathbb{P}_s^{\sigma,\tau^\#}[\Pi_1 \text{ is infinite, and } \Pi_1 \in h_1(\mathbf{AS})^\omega],$$

can be proved by induction on the length of h_1, using the definition of σ_1. That $\mathfrak{E}$ is closed under countable monotone unions and intersections it is easy to see. This concludes the proof of Claim 2.

For ß $\in \{1,2\}$ we define the events

$$E_i = \{\Pi_i \text{ is infinite and reaches s infinitely often}\},$$
$$F_i = E_i \cap \{f(\Pi_i) \leq val(\mathcal{G}_i, s) + \epsilon\}.$$

Fix σ be a strategy for Max. From Claim 2, let σ_1 be a strategy in $\mathcal{G}_1$ such that (38) holds. From Claim 1 and the definition of the strategies,

$$\mathbb{P}_s^{\sigma_1,\tau_1^\#}[\{f > val(\mathcal{G}_1, s) + \epsilon, \text{ and } s \text{ is reached infinitely often}\}] = 0.$$

Now Claim 2 implies

$$\mathbb{P}_s^{\sigma,\tau^\#}[\{f(\Pi_1) > val(\mathcal{G}_1, s) + \epsilon\} \cap E_1] = 0,$$

and symmetrically

$$\mathbb{P}_s^{\sigma,\tau^\#}[\{f(\Pi_2) > val(\mathcal{G}_2, s) + \epsilon\} \cap E_2] = 0.$$

From here it follows that

$$\mathbb{P}_s^{\sigma,\tau^\#}[F_1 \cap F_2 \mid E_1 \cap E_2] = 1.$$

In the end, using the assumption that f is submixing, from (37) we have

$$\mathbb{P}_s^{\sigma,\tau^\#}[\{f \leq \max\left(val(\mathcal{G}_1, s), val(\mathcal{G}_2, s)\right) + \epsilon\} \mid E_1 \cap E_2] = 1.$$

This concludes the proof to the proposition and to Theorem 2.37. ∎

2.6 EXAMPLES

We have given a sufficient condition in Theorem 2.37 for a game to be half-positional. The condition is that the payoff function is Borel-measurable and bounded (so that the game is determined), that it is shift-invariant and submixing (so that Max has an optimal strategy that is stationary and pure). When enumerating all four conditions one might be lead to believe that this is quite restrictive. We will give arguments that point out to the contrary, in the form of examples of payoff functions that fulfill the condition of being shift-invariant and submixing. Almost all of the results that will follow are already known, some are classical, our intention is to understand what is common to them and to give a unified proof.

Throughout this chapter it was Max who was the protagonist — the player that interests us, but one might ask what about optimal strategies of Min, are

they stationary and pure? The quick answer is to just take the payoff function $-f$, if it is submixing and shift-invariant then Min also possesses optimal strategies that are stationary and pure. Equivalently is the payoff function f shift-invariant and such that for all infinite sequences of colors $h, h_1, h_2 \in \mathbf{C}^\omega$ where h is a shuffle of h_1 and h_2,

$$f(h) \geq \min\big(f(h_1), f(h_2)\big). \tag{39}$$

So for some of the payoff functions Theorem 2.37 can be used to show that the game is positional.

All of the payoff functions that we will treat in what follows are obviously shift-invariant, so the only question will be whether they are submixing or not.

Parity payoff. We introduced the parity payoff earlier in Section 2.1. Her the set of colors is some finite set of natural numbers $\mathbf{C}$ and the payoff is defined for sequences $c_0c_1\cdots \in \mathbf{C}^\omega$ as

$$f_{parity}(c_0c_1\cdots) = \begin{cases} 1 & \text{if } \limsup_n c_n \text{ is even} \\ 0 & \text{otherwise,} \end{cases}$$

i.e. the highest color that is seen infinitely often is even. Let $h, h_1, h_2 \in \mathbf{C}^\omega$ such that h is a shuffle of h_1 and h_2. If the highest color that appears infinitely often in h is even (so $f_{parity}(h) = 1$) then this color must appear infinitely often in at least one of h_i and it is the highest color appearing there as well, so for an $i \in \{1,2\}$, $f_{parity}(h_i) = 1$. Therefore f_{parity} is submixing. It also fulfills (39). If the highest color seen infinitely often in h is odd, then it must appear infinitely often in one of h_i where it is the highest color as well. Consequently Theorem 2.37 proves that in perfect-information two player stochastic games with the parity payoff both players Max and Min have optimal strategies that are stationary and pure.

Limsup and liminf payoff. As one can guess from the name, the limsup and liminf payoff functions compute the limsup and liminf of infinite sequences of real numbers, respectively. To see that they are submixing and even fulfill (39) is immediate, since for all $h, h_1, h_2 \in \mathbf{C}^\omega$ such that h is a shuffle of h_1 and h_2,

$$f_{\limsup}(h) = \max\big(f_{\limsup}(h_1), f_{\limsup}(h_2)\big),$$

and

$$f_{\liminf}(h) = \min\big(f_{\liminf}(h_1), f_{\liminf}(h_2)\big).$$

Another proof of the positionality of $f_{\limsup}$ can be found in [Maitra and Sudderth, 2012].

Mean payoff. In the mean payoff, the colors are real numbers, and the function calculated is the average of the colors over the infinite run. Since we allow

players to have infinite memories, the average may not converge. Therefore we had two versions of the mean payoff, for all $c_0c_1\cdots \in \mathbf{C}^\omega$,

$$f_{\overline{mean}}(c_0c_1\cdots) = \limsup_{n\to\infty} \frac{1}{n+1}\sum_{i=0}^{n} c_i,$$

and

$$f_{\underline{mean}}(c_0c_1\cdots) = \liminf_{n\to\infty} \frac{1}{n+1}\sum_{i=0}^{n} c_i.$$

We can show that $f_{\overline{mean}}$ is submixing in the following way. We reproduce the proof from [Gimbert, 2007]. Let $h, h_1, h_2 \in \mathbf{C}^\omega$ such that h is a shuffle of h_1 and h_2. Let f be the mean of a finite word of colors, then for all $n \in \mathbb{N}$, we factor the word of the first n colors into the words d_1 and d_2 depending on whether they appear in h_1 or h_2 such that

$$f(c_0\cdots c_{n-1}) = \frac{|d_1|}{n}\Big(\frac{1}{|d_1|}\sum_{i\in d_1} c_i\Big) + \frac{|d_2|}{n}\Big(\frac{1}{|d_2|}\sum_{i\in d_2} c_i\Big).$$

Let m be the maximum of the two expressions appearing in parentheses above, then

$$f(c_0\cdots c_{n-1}) \le \frac{|d_1|}{n}m + \frac{|d_2|}{n}m = m.$$

Since this holds for every $n \in \mathbb{N}$, we conclude that $f_{\overline{mean}}$ is submixing. Note however that it does not fulfill (39). This is because for sequences $a = (a_n)_{n\in\mathbb{N}}$, $b = (b_n)_{n\in\mathbb{N}}$, in general it is not true that

$$\limsup_n \max(a_n, b_n) \le \max\big(\limsup_n a_n, \limsup_n b_n\big).$$

The situation for $f_{\underline{mean}}$ is symmetrical, *i.e.* (39) holds, but it is not submixing. To summarize Theorem 2.37 implies that in a game with $f_{\overline{mean}}$, Max has an optimal strategy that is both stationary and pure, whereas in a game $f_{\underline{mean}}$ it implies that it is Min who has an optimal strategy that is stationary and pure. We know that in both games, it is both players that possess such optimal strategies, see Section 2.1 and *e.g.* [Gimbert and Zielonka, 2009]. This is not captured by Theorem 2.37. Nevertheless this theorem implies that even for Min in $f_{\overline{mean}}$ and for Max in $f_{\underline{mean}}$, among all finite-memory strategies, the best one is stationary and pure. This is because with finite memory strategies the sequence of averages converges and these two payoff functions coincide.

Positive average payoff. Here, Max wants to keep the average of the rewards to be nonzero. For all $h \in \mathbf{C}^\omega$,

$$f_{posavg}(h) = 1 \iff f_{\overline{mean}}(h) \ge 0.$$

Observe that this payoff function, in fact, gives a very different game from the mean payoff function. For example, with the positive average payoff, Max

prefers to have larger chance of getting a trajectory with a small but positive mean, which is not the case for the mean payoff.

The positionality of games with positive average payoff does not follow easily from the positionality of games with mean payoff. But the positive average payoff function is submixing thanks to the fact that $f_{\overline{mean}}$ is submixing. In fact we can prove that every increasing function that is composed with a submixing function is itself submixing.

Generalized mean payoff. This payoff function was introduced in [Chatterjee *et al.*, 2010]. The colors are an element of $\mathbb{R}^k$ for some fixed $k \in \mathbb{N}$. For every component we compute the mean payoff, and Max wins if in all the components the average is nonzero. In [Chatterjee *et al.*, 2010], the authors show that when computing the average with $f_{\overline{mean}}$ or $f_{\underline{mean}}$, Max may require infinite memory. In the latter case the generalized mean payoff function fulfills (39), hence Min has an optimal strategy that is stationary and pure.

One can consider an optimistic version of generalized payoff, defined as follows. For Max to win, instead of requiring that in *all* the components the average is nonzero, we require that there *exists* some component that has a nonzero average. When computing the average with $f_{\overline{mean}}$ this function is submixing and therefore Max has an optimal strategy that is stationary and pure.

co-Büchi mean payoff. In this payoff function each state is assigned a reward — a real number, and the colors are elements of $\mathbf{S} \times \mathbb{R}$, pairs of state and reward. Let m be the smallest reward. There is $F \subseteq \mathbf{S}$, a set of distinguished states. Max wins the co-Büchi payoff (f_{cb}) if and only if the set of states that appear infinitely often is a subset of F. The co-Büchi mean payoff is defined as follows. It combines the two payoff functions. For all $(s_0, r_0)(s_1, r_1) \cdots \in (\mathbf{S} \times \mathbb{R})^\omega$, where r_i is the reward on state s_i

$$f_{cbm}\big((s_0, r_0)(s_1, r_1) \cdots\big) = \begin{cases} f_{\overline{mean}}(r_0 r_1 \cdots) & \text{if } f_{cb}(s_0 s_1 \cdots) = 1 \\ m & \text{otherwise.} \end{cases}$$

That this function is submixing follows easily from the fact that $f_{\overline{mean}}$ is submixing and that if the co-Büchi is won by Max in some infinite history h, then it is also won in the histories h_1, h_2, given that h is a shuffle of them.

Another function that we mentioned in Section 2.1 was the discount payoff. Observe that this function is not shift-invariant. Nevertheless a game with the discount payoff can be reduced to a game with mean payoff in such a way that an optimal strategy in the latter is an optimal strategy in the former. See [Gimbert, 2006] for details.

2.6.1 *Comments*

The class of submixing functions have some closure properties, *i.e.* when they are composed with increasing functions they remain submixing, a function defined as the maximum of submixing functions is itself submixing, some linear combinations of submixing functions are submixing etc. For more properties for which the class of submixing functions is closed, consult [Gimbert, 2006; Kopczyński, 2009].

The submixing and shift-invariant is not necessary for a game to be half-positional. Moreover for games with a shift-invariant payoff function, the submixing property is not necessary for the game to be half-positional. An example of this is the game where the colors are integers and Max wins if and only if the $\liminf_n$ of the sum of the first n colors equals $-\infty$, [Brázdil *et al.*, 2010].

In light of the main result of [Gimbert and Zielonka, 2009], which says that if both Max and Min, in one player games (MDPs) possess optimal strategies that are both stationary and pure then the two player game is positional as well, one might wonder whether an analogous result holds for half-positionality. Is it true that if Max possesses optimal strategies that are stationary and pure in all one player games, then the two player game is half-positional. This is not true, as it can be witnessed by the following example.

Let the set of colors be $\mathbf{C} = \{a, b\}$, and define $f : \mathbf{C}^* \to \{0, 1\}$, for all $h \in \mathbf{C}^*$

$$f(h) = \begin{cases} 0 & \text{if } h \text{ and } pab^2ab^4 \cdots \text{ have a common suffix for some } (p \in \mathbf{C}^*) \\ 1 & \text{otherwise.} \end{cases}$$

In a one player game equipped with such a payoff function it is very easy to win as Max, in fact, since the set of states is finite, any stationary and pure strategy is optimal. In the other hand there is a simple two player game where Max can win but not with a stationary and pure strategy. Its arena is depicted in Figure 6.

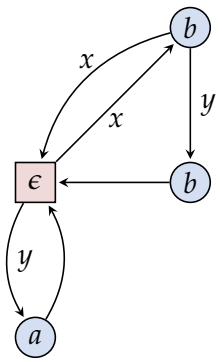


Figure 6: An arena for the game with payoff f

Here the states are labeled by their colors, where ϵ is the empty word. Note that the function c that maps infinite histories to $\mathbf{C}^\omega$ is well-defined because the sole state that is controlled by Min, and labeled by ϵ does not have a self-loop. Assume that Min plays with a stationary and pure strategy, then in the sole state where he has choice he will always choose to play either x or y. In this case Min can create the sequence $ab^2ab^4\cdots$, and win the game with probability 1. Whereas if Max plays *e.g.* uniformly between the actions x and y then Min has no hope to create such a sequence that has so much structure. Max can prevent Min from building such a sequence also by using memory.

2.7 CONCLUSION

In this part of the thesis we have studied two player stochastic games of infinite duration, that are played on finite graphs where both players are perfectly informed. We have tried to answer the question of when the protagonist player Max has an optimal strategy that does not use neither memory nor randomization. We have provided a sufficient condition of when Max has such an optimal strategy. This is a condition on the payoff function, it has to be shift-invariant and submixing. But this condition is not necessary for Max to have a memoryless and deterministic optimal strategy, for example, discounted games are not shift-invariant but they are positional.

One possible research direction is to try to characterize completely when Max has such a simple optimal strategy, in other words to find a condition that is both necessary and sufficient. Alternatively one can ask, what condition is both sufficient and necessary for Max to have an optimal strategy that is only memoryless or only deterministic.

The reason why questions as above are interesting is that ultimately we want to construct the optimal strategy. In this sense, a very important question is what conditions do we need to force on the payoff function such that in the resulting game we are able to effectively construct the optimal strategy of Max?

Part II

GAMES WITH ZERO INFORMATION

3

PROBABILISTIC AUTOMATA

Probabilistic automata are a generalization of deterministic finite state automata, where nondeterminism is resolved by transition probabilities. When a letter is read from a deterministic finite state automaton, it transitions to a new state deterministically, whereas when a probabilistic automaton reads a letter there is a probability distribution according to which the next state is randomly determined. A deterministic automaton either accepts a word or it does not, while a probabilistic automaton accepts a word with some probability. As with other notions of automata, we can consider probabilistic automata over finite words as well as over infinite words. In this chapter we will concentrate only on the former. Questions such as, does there exist a word that is accepted with probability larger than c (emptiness), and whether all words are accepted with probability larger than c (universality), are undecidable, where c is a fixed rational number in $(0, 1)$. The *value 1 problem* has the same fate. This is the question whether for all $\epsilon > 0$ there exists a word that is accepted with probability larger than $1 - \epsilon$. Recently there has been some effort to find interesting and robust classes of automata for which, problems such as the value 1 problem are decidable. Notably *leaktight automata* in [Fijalkow *et al.*, 2012] and *simple automata* in [Chatterjee and Tracol, 2012], for both of which the value 1 problem is decidable. In this chapter we are going to show that simple automata are strictly contained in the class of leaktight automata. In doing so, we will introduce some notions that are used in [Fijalkow *et al.*, 2012] for deciding the value 1 problem for leaktight automata which will be useful for the chapter that follows, where we will lift this decidability result to two player *half-blind* games.

One can view probabilistic automata as a one player game with the winning objective of reaching the final states, where the player has no information about the state of the game. While in the first part of this thesis we considered games with perfect information, now we will consider the other extreme: games where the protagonist has no information.

3.1 OVERVIEW

In deterministic finite automata the transition function maps pairs of state, and letter of the alphabet, $\mathbf{S} \times \mathbf{A}$, to states $\mathbf{S}$. Whereas in probabilistic automata the transition function maps pairs of state, and letter of the alphabet, $\mathbf{S} \times \mathbf{A}$ to probability distributions on states $\Delta(\mathbf{S})$.

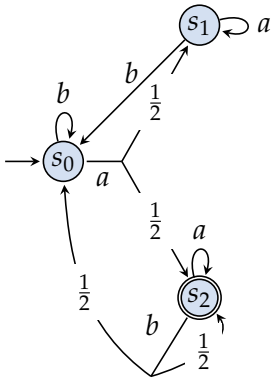


Figure 7: A probabilistic automaton

In the example in Figure 7 the initial state of the automaton is s_0, the final state is s_2. When the letter a is read in state s_0 the automaton moves with equal probability to s_1 or s_2. At this point, if Figure 7 was an arena of a Markov decision process (MDP) we could base the decision on what action to take next on whether the current state is s_1 or s_2. This is not the case in a probabilistic automaton, we have to *blindly* choose whether a or b is the next letter of the word. Because of this, a probabilistic automaton can be seen as a one player game with zero information. In other words, in an MDP we are allowed to play according to general strategies, while in a probabilistic finite automaton, we search among particular and very simple strategies, *i.e.* finite words of actions.

Probabilistic automata were first introduced by Rabin in [Rabin, 1963]. Even though most problems for probabilistic automata are undecidable it has a rich literature. We will briefly discuss a portion of said literature in this section. The interested reader is referred to the book-length treatment [Paz, 1971] and the survey [Bukharaev, 1980].

In [Rabin, 1963], the language theoretic point of view is taken, with questions such as, are stochastic languages more expressive than other formal languages *e.g.* regular languages. A stochastic language of a given probabilistic automaton with respect to some cut-point $0 \leq c < 1$ is defined as the set of words that are accepted by the automaton with probability strictly larger than c. Rabin proves that there are uncountably many stochastic languages, hence there are some stochastic languages that are not regular.

Theorem 3.1 ([Rabin, 1963]).
There exists a probabilistic automaton with a cut-point c such that the language of words that are accepted with probability strictly larger than c is not regular.

Moreover he claims that it is possible to construct a particular language that is not regular that is accepted by a probabilistic automaton, albeit with an irrational cut-point. In [Paz, 1971, Theorem 3.6, page 167] it is proved that there exists a three state probabilistic automaton with a single letter alphabet and a rational cut-point that defines a stochastic language that is not regular.

On the other hand if the cut-point c is *isolated* then the stochastic language with respect to c is regular. We say that a cut-point c is isolated if there exists some $\epsilon > 0$ such that there is no word that is accepted with probability in the ϵ-neighborhood of c, we call ϵ the *degree of isolation*. Moreover it is shown that there is an upper bound as a (effective) function of ϵ on the number of states of the deterministic automaton that recognizes the same language as the probabilistic automaton with an isolated cut-point.

Theorem 3.2 ([Rabin, 1963]).
For all probabilistic automata $\mathbb{A}$ and cut-points c, if c is isolated with degree of isolation ϵ, the stochastic language of $\mathbb{A}$ with respect to the cut-point c is regular and the deterministic finite automaton that recognizes it has at most $(1+\epsilon^{-1})^{n-1}$ states where n is the number of states of $\mathbb{A}$.

Rabin further studies a subclass of automata called *actual automata* which enjoys good properties. These are automata where the support of every probability distribution in the transition table is the full set of states $\mathbf{S}$, *i.e.* there is nonzero probability to go from any state to any other state with any letter of the alphabet.

Emptiness is undecidable for probabilistic automata [Paz, 1971]. Emptiness is the question of whether there exists a word that is accepted with probability strictly larger than c. If we replace "strictly larger" by larger, smaller, equal, strictly smaller, the question remains undecidable. An even stronger statement holds in fact

Theorem 3.3 ([Condon and Lipton, 1989]).
Given a probabilistic automaton $\mathbb{A}$ and some rational $\epsilon > 0$ there does not exist any algorithm such that

- if there exists some word that is accepted by $\mathbb{A}$ with probability strictly larger than $1 - \epsilon$ then it replies **yes**,
- if all words are accepted with probability smaller than ϵ it replies **no**.

Moreover emptiness is undecidable even for automata of fixed size [Blondel and Canterini, 2003].

In contrast to this, observe that from Theorem 3.2 if the cut-point is isolated and we know the degree of isolation then the emptiness problem is decidable because we have a bound on the length of the words that are sufficient to check. The decidability of the emptiness problem when the cut-point is isolated, but the degree of isolation is not known, is an open question.

As a consequence we see that whether the cut-point is isolated or not plays an important role in the difficulty of decision problems such as emptiness in probabilistic automata; but the isolation problem itself is not decidable either.

Theorem 3.4 ([Bertoni, 1975]).
Given a probabilistic automaton $\mathbb{A}$ and a rational cut-point $c \in (0, 1)$ the problem of whether c is isolated (*i.e.* does there exists some ϵ such that no word is accepted by $\mathbb{A}$ with probability in $[c - \epsilon, c + \epsilon]$) is undecidable.

Notice that in the theorem above Bertoni leaves open the question when the cut-point is equal to 1 or 0. In [Gimbert and Oualhadj, 2010] it is shown that it remains undecidable for $c = 1$ and $c = 0$. The former is called the *value 1* problem and is the subject of this chapter.

Problem 3.5 (The value 1 problem).
Given a probabilistic automaton $\mathbb{A}$, decide whether for all $\epsilon > 0$ there exists a word that is accepted by $\mathbb{A}$ with probability larger than $1 - \epsilon$.

While the problem above is undecidable in general there have recently been some work on identifying interesting classes of automata for which it is decidable. Before introducing them, let us survey a few more results on probabilistic automata.

The equivalence problem is decidable. We are given two probabilistic automata and we want to decide whether for all words w the probability of accepting w from both automata is equal. The decidability of this problem follows from the work of Schützenberger on minimizing automata [Schützenberger, 1961]. Another algorithm for this problem can be found in [Tzeng, 1992].

Probabilistic automata with a unary alphabet are studied in [Chadha *et al.*, 2014]. These are equivalent to Markov Chains. The authors show that the isolation problem (the problem of deciding whether the a cut-point is isolated) is $coNP$-complete for cutpoints $c \in \{0, 1\}$ and it is in $PSPACE$ for cutpoints $c \in (0, 1)$.

In [Chadha *et al.*, 2013] it is demonstrated that the exact level of undecidability of the isolation problem is Σ_2^0-complete. Then they define a class of automata called the eventually weakly ergodic automata for which the emptiness problem is decidable when the cut-point is isolated, even if the degree of isolation is not known.

While we will concentrate on automata on finite words in this thesis, probabilistic automata on infinite words are of interest as well, especially considering their possible applications. Probabilistic Büchi automata were introduced first in [Baier and Grosser, 2005]. They are probabilistic automata on infinite words with the Büchi acceptance condition, *i.e.* at least one of the infinitely occurring states belongs to the set of final states. They are then compared in terms of language expressiveness with nondeterministic ω-automata and shown to be more expressive, as in the case of finite word automata. Two semantics are considered, the nonzero semantics where we define the language accepted by the automaton to be the set of infinite words whose probability of acceptance is nonzero, and the almost sure semantics. In [Baier *et al.*, 2008] the authors show that all interesting problems such as emptiness universality etc. are undecidable for the nonzero semantics; however the almost sure semantics has a decidable emptiness problem. The exact complexity of these problems is closed in [Chadha *et al.*, 2009]. It is shown that emptiness and universality for the nonzero semantics are Σ_2^0-complete and for the almost sure semantics they are $PSPACE$-complete. Further semantics and acceptance conditions are considered in [Chatterjee and Henzinger, 2010]. Except the nonzero and almost sure semantics, the limit semantics is considered as well (the value 1 problem introduced above), and different acceptance conditions such as safety, reachability, Büchi, co-Büchi etc.

The decidability results for the positive semantics and co-Büchi, as well as almost sure semantics and Büchi were later generalized to game of partial information in [Bertrand *et al.*, 2009], and [Gripon and Serre, 2009].

In [Korthikanti *et al.*, 2010] the probability distributions on the set of states are labeled by colors picked from a finite set. As an infinite word is read by a probabilistic automaton it produces an infinite word of colors. The question is can one decide whether the intersection of the language of infinite words of colors (for some *linear* labeling) and of some regular language is nonempty (the model checking problem). In general it is undecidable because of the undecidability of the emptiness mentioned above. But for a special class of contracting automata and strategies it is decidable. A similar point of view is taken in [Chadha *et al.*, 2011] where it is shown that for automata that have a unique and compact invariant set of distributions which are called *semi-regular* it is decidable to model check any ω-regular property. As a corollary the authors prove that for semi-regular probabilistic automata on finite words with an isolated cut-point the emptiness problem is decidable, even if the degree of isolation is not known.

Another decidable problem for probabilistic automata on infinite words is that of synchronization [Doyen *et al.*, 2011]. This is the problem of deciding whether there exists some infinite word such that the highest probability in the sequence of distributions that is generated by it tends to 1, *i.e.* in the limit it behaves like a deterministic automaton.

We come back on probabilistic automata on finite words and the value 1 problem. Gimbert and Oualhadj prove that the value 1 problem for probabilistic automata on finite words is undecidable [Gimbert and Oualhadj, 2010], this problem was left open by Bertoni. The proof reduces the value 1 problem to the *equality problem*, this is the problem of deciding whether there exists some word with acceptance probability exactly equal to $\frac{1}{2}$. The equality problem in turn can be reduced to Post's correspondence problem. Furthermore the authors introduce a class of automata called #-*acyclic automata* for which the value 1 problem is decidable. The decision procedure constructs the *support graph* of the automaton. This is a graph whose vertices are the powerset of states, and we add an edge from the vertex $S_1 \subseteq \mathbf{S}$ to the vertex $S_2 \subseteq \mathbf{S}$ if it is possible for the automaton to transition from a distribution with support S_1 to a distribution with support S_2. The transitions can happen from reading a finite word, but it also takes into account the fact that by repeating some word we can decrease the distribution on some states arbitrarily: this is what #-operator captures. If this support graph does not have any cycles, except for simple loops, we say that the corresponding automaton is #-cyclic.

Another class of probabilistic automata was defined in [Chadha *et al.*, 2009], the class of *hierarchical probabilistic Büchi automata*. These are automata that have a hierarchical structure: the set of states is partitioned into levels and at

each level, from every state there can be at most one transition that goes to some state in the same level while all others go to states at higher levels. These automata with respect to the nonzero semantics (for the probabilistic Büchi automata) define exactly the class of ω-regular languages, and for the almost sure semantics they define the class of ω languages that are recognized by deterministic Büchi automata. Moreover the universality and emptiness problems are tractable for both semantics in this class of automata.

The subject of this chapter is the comparison between two classes of automata: the leaktight automata and the simple automata.

The class of *leaktight* probabilistic finite automata is introduced in [Fijalkow *et al.*, 2012]. This class of automata has a decidable value 1 problem. The authors show that the automaton can be abstracted with a finite monoid called *the Markov monoid* which has sufficient information for deciding whether the automaton has value 1 or not — given that the automaton is leaktight. Roughly speaking, a leak occurs in a probabilistic finite automaton if there is some communication between different recurrence classes that we attain in the limit of a sequence of finite words. This will be defined precisely in the section that follows, and it will be discussed in more detail in the next chapter.

The Markov monoid abstracts away the exact transition probabilities, the only information it retains is whether the probability transition is zero or nonzero. We can partition the set of all probabilistic automata into sets for which all zero transition probabilities coincide, *e.g.* for all automata on such a set with letter a there is nonzero probability to go from state s to state t. We can associate to any such set a *numberless* automaton. Since the Markov monoid is the same for every probabilistic automaton associated to the same numberless automaton this begs the question whether the information that the Markov monoid is capturing is sufficient to decide whether, for example, there exists an automaton in the set that has value 1. This is not the case. In [Fijalkow *et al.*, 2014], it is proved that that for all probabilistic automata $\mathbb{A}$ one can construct a numberless automaton such that, $\mathbb{A}$ has value 1 if and only if all automata associated to the numberless automaton have value 1, which in turn is equivalent to: there exists some automaton associated to the numberless automaton that has value 1.

Probabilistic automata on infinite words with parity conditions are considered in [Chatterjee and Tracol, 2012], and three semantics, the nonzero, almost sure, and the limit semantics (*i.e.* the value 1 problem). The authors identify the class of *structurally simple probabilistic automata*, which is a syntactic restriction on the support graph that is associated to the automaton. Then it is proved that this class of automata generalizes both #-acyclic and hierarchical automata that we mentioned above.

Interestingly the techniques of [Fijalkow *et al.*, 2012] and [Chatterjee and Tracol, 2012] are quite different. The former uses the theory of finite semi-groups,

in particular Simon's forest factorization theorem [Simon, 1990]. Whereas the latter uses a result from probability theory, namely the decomposition separation theorem [Sonin, 2008]. Given a morphism from the set of finite words over an alphabet to a finite monoid, Simon's theorem shows how to factor (with respect to the morphism) any given word into a tree whose height does not depend on the length of the word but only on the size of the finite monoid. The decomposition separation theorem generalizes a theorem on homogeneous Markov chains, that partitions the set of states into recurrent and transient classes (sometimes called essential and nonessential). The generalization is from homogeneous to nonhomogeneous, and the decomposition is into *jets*.

In this chapter we will report on [Fijalkow *et al.*, 2015], and prove that the class of leaktight automata strictly contains the structurally simple automata, and consequently it is the largest and most robust class of automata for which it is known that the value 1 problem is decidable. The proof demonstrates that the information of whether the automaton is simple or not can be found in the Markov monoid.

The class of leaktight automata and the Markov monoid algorithm are also interesting because they are optimal in some sense [Fijalkow, 2015]. A probabilistic automaton has value 1 if and only if there exists a sequence of finite words that witnesses it, that is the probability of accepting the n-th word tends to 1 as n tends to infinity. Nathanaël Fijalkow demonstrates in [Fijalkow, 2015] that the Markov monoid algorithm replies *yes* if and only if there exists a *polynomial* sequence of words that witnesses the value 1. The set of polynomial sequences of words is attained by taking the set of constant sequences and closing it by concatenation and iteration, where the iteration of a sequence $(u_n)_{n\in\mathbb{N}}$ is defined to be $(u_n^n)_{n\in\mathbb{N}}$. But once we walk away from the polynomial sequences even the following — simpler than the value 1 — problem is undecidable: given a probabilistic automaton, determine whether there exist two finite words u and v such that $\left((uv^n)^{2^n}\right)_n$ witnesses the value 1.

Another point of interest of this class is that it can be lifted to a type of two player games. In the chapter that follows we will show that the maxmin reachability is decidable for leaktight *half-blind* games, where we will be using some of the notions that will be introduced in this chapter, namely the Markov monoid.

A reason for the size of the literature on probabilistic automata, even though most problems are undecidable, is that there are numerous applications from a variety of fields. For example in pharmacokinetics, the way that a drug moves in an organism can be modeled as a probabilistic automaton [Shargel *et al.*, 2007; Korthikanti *et al.*, 2010]. Another area of applications is software verification [Vardi and Wolper, 1986; Baier *et al.*, 2012; Raskin *et al.*, 2007]. Probabilistic automata have also been used in other places in computational biology [Durbin

et al., 1998], in speech processing [Mohri, 1997] etc. Suffice it to say that it is a useful tool for modeling a certain type of probabilistic machines.

3.2 DECIDABLE CLASSES

We will define probabilistic automata on finite words, and then introduce the classes of leaktight and simple automata. In the next section we prove that the latter is a strict subset of the former.

Definition 3.6 (Probabilistic automaton).
A probabilistic automaton $\mathbb{A}$ is a tuple $\mathbb{A} = (\mathbf{S}, \mathbf{A}, s_0, \delta, \mathbf{F})$, where

- $\mathbf{S}$ is a finite set of states,
- $\mathbf{A}$ is a finite alphabet,
- $s_0 \in \mathbf{S}$ is an initial state,
- $\delta : \mathbf{S} \times \mathbf{A} \to \Delta(\mathbf{S})$, is the transition function, and
- $\mathbf{F} \subseteq S$ is a set of final states.

Comparing this definition with Definition 2.1, one can see probabilistic automata as the arena of a one player game. The differences are as follows, here we will be interested uniquely on the reachability of the set of final states as opposed to some general payoff function, the strategies that we are allowed to use are very special: the set of finite words, and the game is of finite (but arbitrary long) duration.

The transition function δ induces a stochastic matrix for every $a \in \mathbf{A}$. Stochastic matrices are matrices whose each line is a probability distribution. In our case they are indexed by elements of $\mathbf{S}$, *e.g.* for the letter $a \in \mathbf{A}$, the stochastic matrix M_a is defined as $M_a(s, s') = \delta(s, a)(s')$, where $s, s' \in \mathbf{S}$. Essentially questions about probabilistic automata are questions over the semi-group that is generated by $\{M_a \mid a \in \mathbf{A}\}$.

Let $w \in \mathbf{A}^*$ and $s, s' \in \mathbf{S}$, we will use the notation

$$\mathbb{P}(s \xrightarrow{w} s'),$$

to mean the probability of going from state s to state s' with the word w. This can be precisely defined inductively on the length of the word, that is for every letter $a \in \mathbf{A}$ and states $s, s' \in \mathbf{S}$, we define

$$\mathbb{P}(s \xrightarrow{a} s') = \delta(s, a)(s'),$$

and for every word $w \in \mathbf{A}^*$, letter $a \in \mathbf{A}$, and states $s, s' \in \mathbf{S}$,

$$\mathbb{P}(s \xrightarrow{wa} s') = \sum_{t \in \mathbf{S}} \mathbb{P}(s \xrightarrow{w} t)\mathbb{P}(t \xrightarrow{a} s').$$

For example the question of whether there exists a word that is accepted with probability strictly larger than $\frac{1}{2}$ is formulated as whether there exists some $w \in \mathbf{A}^*$ such that $\sum_{t \in \mathbf{F}} \mathbb{P}(s_0 \xrightarrow{w} t) > \frac{1}{2}$. We will also use the following notation shorthand, for all $s \in \mathbf{S}$, $\mathbf{S}' \subseteq \mathbf{S}$, and $w \in \mathbf{A}^*$

$$\mathbb{P}(s \xrightarrow{w} \mathbf{S}') = \sum_{s' \in \mathbf{S}'} \mathbb{P}(s \xrightarrow{w} s').$$

In parallel to the two player games that were the subject of the previous chapter, we define the value of a probabilistic automaton as follows.

Definition 3.7 (Value of a probabilistic automaton).
Let $\mathbb{A}(\mathbf{S}, \mathbf{A}, s_0, \delta, \mathbf{F})$ be a probabilistic automaton, then its value is

$$val(\mathbb{A}) = \sup_{w \in \mathbf{A}^*} \mathbb{P}(s_0 \xrightarrow{w} \mathbf{F}).$$

With the definition above, the value 1 problem (Problem 3.5) is the question of whether $val(\mathbb{A}) = 1$.

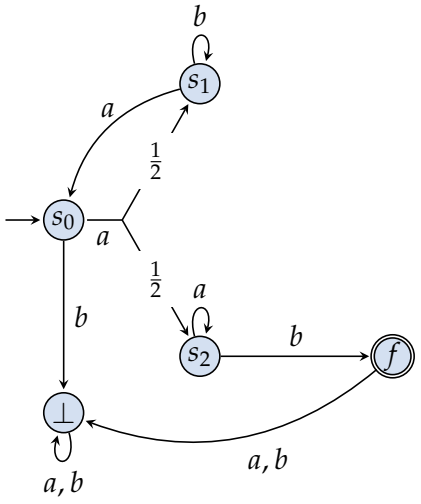


Figure 8: A probabilistic automaton with value 1

Example 6.
In the automaton in Figure 8, the initial state is the state s_0 (this is denoted by the incoming arrow), and the (single) final state is the state f (this is denoted by the double circling). The value of this automaton is at least $\frac{1}{2}$, because we see that the word ab is accepted with probability $\frac{1}{2}$. But we can do better than this. The word $aaab$ is accepted with probability $\frac{3}{4}$. In general we see that the word $(aa)^k ab$ is accepted with probability $1 - \frac{1}{2^k}$. Consequently this automaton has value 1.

If hypothetically we had $\delta(s_2, a)(\bot) = 1$, the value would be $\frac{1}{2}$, since once we reach the sink state $\bot$ we are trapped there and we have no hope of reaching the final state, hence after playing an a it is best to play a b and then to stop.△

3.2.1 *Leaktight automata*

We define first the class of leaktight automata, introduced in [Fijalkow *et al.*, 2012]. The proof of decidability is algebraic in nature and it uses tools from the theory of finite semi-groups.

We mentioned above that for every letter $a \in \mathbf{A}$ the transition function δ induces a $|\mathbf{S}| \times |\mathbf{S}|$ stochastic matrix, which we denoted by M_a. It is not hard to see that for a word $w = a_1 a_2 \cdots a_n \in \mathbf{A}^*$, and states $s, s' \in \mathbf{S}$,

$$\mathbb{P}(s \xrightarrow{w} s') = (M_{a_1} \cdots M_{a_n})(s, s'),$$

that is, the entry (s, s') of the product $M_{a_1} \cdots M_{a_n}$ is the probability to go from state s to state s' with the word $w = a_1 \cdots a_n$.

The set of all matrices generated by closing $\{M_a \mid a \in \mathbf{A}\}$ under taking products is in general infinite. Therefore there is no hope to compute all of it.

The idea of the Markov monoid algorithm is to abstract away the probabilities of the stochastic matrices and to keep only the information of whether the transition is zero or not, in this way constructing a finite object, a monoid, which will have the relevant information for deciding the value 1 problem, in the subset of leaktight automata. In other words the stochastic matrix is abstracted by a binary one, *i.e.* the matrix M_a is abstracted by the binary matrix B_a whose entries are 1 if and only if the corresponding entry in M_a is nonzero. That is, for all $s, s' \in \mathbf{S}$,

$$B_a(s, s') = 1 \iff M_a(s, s') > 0. \tag{40}$$

The elements of the Markov monoid are binary matrices that are indexed by a pair of states. The operation is matrix product over the boolean semi-ring, in other words, given two binary matrices B and B', and the pair of states $s, s' \in \mathbf{S}$, we have $(BB')(s, s') = 1$ if and only if there exists some state $t \in \mathbf{S}$ such that $B(s, t) = 1$ and $B'(t, s') = 1$, where by BB' we have denoted the product of the two matrices.

A subset of binary matrices together with the product operation described above, and the unit matrix already forms a monoid, but this is not the Markov monoid, in particular it does not have any information about the possible states that can be reached by iterating some word. To illustrate this let us go back to the example in Figure 8.

Example 7.
The monoid in question is the one that is generated by B_a and B_b and has the unit matrix, where

$$B_a = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}, \quad B_b = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}.$$

The states have been ordered as $s_0 < s_1 < s_2 < \perp < f$, *i.e.* the $(0,0)$ entry corresponds to (s_0, s_0). One can quickly see that if we close the set $\{B_a, B_b\}$ by taking products, none of the elements of this closure will have the first row equal to $(0,0,0,0,1)$. This means that there does not exist any word $w \in \mathbf{A}^*$ such that $\mathbb{P}(s_0 \xrightarrow{w} f) = 1$. Nevertheless the example in Figure 8 has value 1.△

In order to account for the fact that in some cases we are able to decrease the probability to be in some state (or increase) arbitrarily (as in the case of the states s_0 and s_1 in the example in Figure 8), another *unary* operation is added, called the *iteration* (sometimes it is called the stabilization operation in the literature). The Markov monoid is a *stabilization monoid*. These are (usually finite) monoids, that are equipped with unary operation (denoted by #) that maps idempotent[1] elements of the monoid to idempotents. Stabilization monoids were used by Simon in [Simon, 1994] and then later by Kirsten in [Kirsten, 2004] for his algorithm for the famous star height problem in automata and formal language theory; and Colcombet gave the general definition of stabilization monoids in [Colcombet, 2009] and used them as important tools for the theory of cost functions. A proof that the Markov monoid is a stabilization monoid, can be found in [Fijalkow *et al.*, 2012].

To motivate the definition of the iteration operation in the context of the Markov monoid and probabilistic automata, let us consider finite space and discrete time, homogeneous Markov chains. Such a chain can be characterized by a single stochastic matrix M. Say that M is a $|\mathbf{S}| \times |\mathbf{S}|$ matrix, for the sake of the illustration. If for all $s, s' \in \mathbf{S}$,

$$M(s,s') > 0 \iff M^2(s,s') > 0, \tag{41}$$

1 elements M such that $M^2 = M$

then we say that the Markov chain characterized by M is *idempotent* and we can partition the set of states into *recurrent* and *transient* states. Given that the Markov chain is recurrent, we say that a state $r \in \mathbf{S}$ is recurrent if and only if for all $r' \in \mathbf{S}$, $M(r, r') > 0$ implies that $M(r', r) > 0$. A state is transient if it is not recurrent. So a state r is recurrent if for all states r', if there is nonzero probability of reaching r' from r, then there is nonzero probability of coming back, *i.e.* of reaching r from r'. This is not the usual definition of recurrence for general Markov chains, but it coincides with the usual one if the Markov chain is idempotent. It is only for idempotent Markov chains that we will talk about recurrence and transience.

A fundamental theorem for Markov chains says that the more steps the chain is run, the less there is chance to be in a transient state. In other words for all states $s, t \in \mathbf{S}$ if t is transient then $\liminf_n M^n(s, t) = 0$. Intuitively this is because the transient state gives away some probability that is never returned.

Example 8.
Observe that the stochastic matrix M_a^2 from the example in Figure 8 verifies property (41). Moreover the states s_0 and s_1 are transient (with respect to the chain associated with M_a^2) because both $M_a^2(s_0, s_2)$ and $M_a^2(s_1, s_2)$ are nonzero, whereas both $M_a^2(s_2, s_0)$ and $M_a^2(s_2, s_1)$ are. As a consequence, the first row in the sequence $(M_a^{2n})_n$ will tend to $(0, 0, 1, 0, 0)$. △

Now one can imagine the right definition of the iteration operator #, after defining the notions of recurrent, transient etc, for the binary matrices, which we do now. We assume that all matrices are indexed by the states $\mathbf{S}$.

Definition 3.8 (Product operator for the Markov monoid).
The product of the binary matrices B and B', denoted BB' is defined for all $s, s' \in \mathbf{S}$, as

$$BB'(s, s') = 1 \iff \exists\, t \in \mathbf{S},\ B(s, t) = 1 \text{ and } B'(t, s') = 1.$$

Definition 3.9 (Idempotent binary matrix).
A binary matrix B is idempotent if and only if $B^2 = B$.

Definition 3.10 (B-recurrence).
Let B be an idempotent binary matrix, and $r \in \mathbf{S}$. We say that r is *B-recurrent* if and only if for all $r' \in \mathbf{S}$,

$$B(r, r') = 1 \implies B(r', r) = 1.$$

If a state is not recurrent we say that it is *transient*.

Definition 3.11 (Iteration operator for the Markov monoid).
Let B be an idempotent binary matrix. We define the *iteration* of B, denoted by $B^{\#}$, for all $s, t \in \mathbf{S}$ as

$$B^{\#}(s, t) = 1 \iff B(s, t) = 1 \text{ and } t \text{ is } B\text{-recurrent}.$$

In other words, the iteration operation deletes incoming edges to transient states.

Having defined both of the operators now we are ready for the definition of the Markov monoid that is associated to a probabilistic automaton. For a set of binary matrices X denote by $\langle X \rangle$, the smallest set that contains X and is closed under taking products and iterations.

Definition 3.12 (Markov monoid).
The Markov monoid $\mathcal{M}$ is defined as

$$\mathcal{M} = \langle \{ B_a \mid a \in \mathbf{A} \} \cup \{ I \} \rangle,$$

where the elements B_a are as in (40), and I is the unit matrix.

Example 9.
We go back yet again to the example in Figure 8, and consider the element of the monoid $B_a^2 = B_{aa}$. It is an idempotent element therefore $B_{aa}^{\#}$ is well-defined. Calculating it yields

$$B_{aa}^{\#} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}.$$

This captures the fact that by repeating the word aa many times we can increase the chance of reaching s_2 from s_0 to 1. △

An interesting fact comes up from Definition 3.12. As mentioned before, when we abstract the matrices M_a by the binary matrices and generators of the Markov monoid B_a, we throw away the probabilities and keep only the qualitative properties of the automaton. The exact transition probabilities are obviously important for the problem of emptiness among others, but this raises the question whether they are important for the value 1 problem as well. Unsurprisingly, they are. In [Gimbert and Oualhadj, 2010] the authors demonstrate an example of an automaton that has value 1 if and only if a particular transition probability is strictly larger than $\frac{1}{2}$. In fact they show undecidability of the value 1 problem by reducing it to the emptiness problem[2], where that critical transition is replaced by an automaton. But as we shall see later, for leaktight automata, the qualitative part is sufficient to decide the value 1 problem.

The *Markov monoid algorithm* simply constructs $\mathcal{M}$ associated to a given probabilistic automaton. This object is finite hence such a procedure always terminates. Even though finite, the size of the monoid can be exponentially large in the number of states. But it is not necessary to compute the whole monoid because we are interested in particular elements, called the *value 1 witness*. This element can be guessed, and shown that the Markov monoid algorithm has *PSPACE* upper bound. This bound is tight [Fijalkow *et al.*, 2012].

Definition 3.13 (Value 1 witness).
Let $\mathcal{M}$ be the Markov monoid associated to a probabilistic automaton with initial state s_0 and final states $\mathbf{F}$, then $B \in \mathcal{M}$ is a *value 1 witness* if and only if for all $s \in \mathbf{S}$

$$B(s_0, s) = 1 \implies s \in \mathbf{F}.$$

2 consequently emptiness is reduced to the problem of equality

Example 10.
Computing the product $B^{\#}_{aa}B_b$ in the example in Figure 8 yields

$$B^{\#}_{aa}B_b = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix},$$

which is a value 1 witness. △

The algorithm constructs $\mathcal{M}$ and then searches for a value 1 witness, once it is found, it returns *yes*, if the monoid does not have a value 1 witness then it returns no. The algorithm does not return false positives.

Theorem 3.14 ([Fijalkow *et al.*, 2012]).
Let $\mathbb{A}$ be a probabilistic automaton and $\mathcal{M}$ the Markov monoid associated to it. If $\mathcal{M}$ has a value 1 witness then $val(\mathbb{A}) = 1$.

The converse does not hold in general of course, since the value 1 problem is undecidable; but it does if the automaton is of a special kind: a leaktight automaton. Here we will only give the definition of leaktight automata without any intuition or further investigation. We will postpone this to the next chapter which also deals with leaks.

The (syntactic) definition of leaktight automata relies on the *extended Markov monoid*. Think of this monoid as a Markov monoid where we remember the edges that were deleted from the iteration operation by keeping pairs of binary matrices, in such a way that the right component is not modified by iteration.

Definition 3.15 (Extended Markov monoid).
The extended Markov monoid, denoted by $\widetilde{\mathcal{M}}$, is the smallest set that is closed under taking product and iteration and contains all the elements

$$\{(B_a, B_a) \mid a \in \mathbf{A}\} \cup \{(I, I)\},$$

where B_a are defined as in (40), whereas the product and iteration are defined as

$$(B, \widetilde{B}) \cdot (C, \widetilde{C}) = (BC, \widetilde{B}\widetilde{C})$$
$$(D, \widetilde{D})^{\#} = (D^{\#}, \widetilde{D}),$$

where $(D, \widetilde{D})$ is idempotent, *i.e.* $(D, \widetilde{D})^2 = (D, \widetilde{D})$.

The extended Markov monoid can be computed the same way that the Markov monoid is.

Roughly speaking, leaks complicate calculations necessary for finding out whether the automaton has value 1, since convergence speeds start to matter. We will identify leaks as elements of the extended Markov monoid that have certain properties. Indeed we will see that the (extended) Markov monoid, has more information about the automaton than whether it has value 1 or not. By calculating the extended Markov monoid we can also decide whether the automaton is leaktight or not, as well as if it is not a *simple automaton*. First let us define leaks and leaktight automata.

Definition 3.16 (Leaktight automata).
Let $\mathbb{A}$ be a probabilistic automaton and $\widetilde{\mathcal{M}}$ the extended Markov monoid that is associated to it. An idempotent element $(B, \widetilde{B}) \in \widetilde{\mathcal{M}}$ is called a *leak* if there exists $r, r' \in \mathbf{S}$ such that

- r and r' are B-recurrent,
- $B(r, r') = 0$,
- $\widetilde{B}(r, r') = 1$.

If $\widetilde{\mathcal{M}}$ does not contain a leak then we say that $\mathbb{A}$ is leaktight.

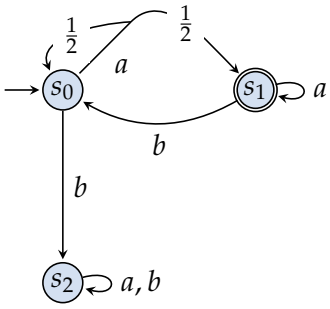


Figure 9: A probabilistic automaton that is not leaktight

Example 11.
Consider the automaton in Figure 9. Calculating the element of the extended Markov monoid

$$(B_a, B_a)^{\#}(B_b, B_b),$$

we see that it is a leak. This is because the states s_0 and s_2 are $B_a^{\#}B_b$-recurrent, $B_a^{\#}B_b(s_0, s_2) = 0$ and $B_aB_b(s_0, s_2) = 1$. △

Indeed if one investigates the asymptotic behavior of the sequence of stochastic matrices $(M_a^n M_b)_{n\in\mathbb{N}}$, one can conclude that it already depends on the exact transition probabilities of the letter a from state s_0. We will come back to this, and other examples of leaks in the next chapter.

3.2.2 *Simple automata*

Let us now turn our attention to the simple automata described in [Chatterjee and Tracol, 2012]. The authors of this paper consider probabilistic automata on infinite words. An infinite word $w \in \mathbf{A}^\omega$ induces a stochastic process on $\mathbf{S}$, that is a sequence of random variables $(X_n)_{n\in\mathbb{N}}$ that takes values in $\mathbf{S}$. The central notion of this paper is that of a *simple process*. A stochastic process induced by some infinite word is simple if it has a particular structure that is inspired by a deep result in probability theory, called the *decomposition-separation theorem*, see [Sonin, 2008] and the references therein. Then it is proved that for words that induce simple processes for the almost sure, the positive semantics, and parity conditions, emptiness is in $PSPACE$. Furthermore the words that are eventually repeating always induce simple process.

A simple automaton is defined as an automaton where all words induce a simple process. *Structurally simple automata* are a subclass of automata that are provably simple (but not all simple automata are structurally simple). It is demonstrated that for structurally simple automata the value 1 problem is decidable and in $EXPSPACE$. They also subsume hierarchical automata and #-acyclic automata.

The decomposition separation theorem decomposes the set of infinite sequences of states into *jets*.

Definition 3.17 (Jets).
A jet $J = (J_n)_{n\in\mathbb{N}}$ is a sequence of subsets of $\mathbf{S}$, *i.e.* for all $n \in \mathbb{N}$, $J_n \subseteq \mathbf{S}$. We say that a tuple of jets $(J^1, \dots, J^k)$ is a decomposition of $\mathbf{S}^\omega$ (the set of all infinite sequences of states) if for all $n \in \mathbb{N}$, $(J_n^1, \dots, J_n^k)$ is a partition of $\mathbf{S}$.

The classical result for homogeneous Markov chains partitions the state space $\mathbf{S}$ into the transient states, and recurrence classes (sets of recurrent states that communicate with each other). Then no matter from what state we start the

chain will enter one of the recurrence classes with probability 1 and stay there forever. It is surprising that a similar result holds when the chain is nonhomogeneous as well.

Note that an infinite word in a probabilistic automaton induces a nonhomogeneous Markov chain where the transition probability at time n depends on the n-th letter of the word. The decomposition separation theorem says that for all nonhomogeneous Markov chains, there exists a decomposition of $\mathbf{S}^\omega$ into jets $(J^0, \ldots, J^k)$ such that with probability 1 we enter one of the jets $J^k, \ldots, J^k$ and stay there forever. Where the jet J^0 plays a role analogous to transient states.

In the case of homogeneous Markov chains we partition the state space, whereas for nonhomogeneous Markov chains we partition the "space time". We have to take time into consideration as well since the chain being nonhomogeneous implies that the transition probabilities change as a function of time.

The idea of a simple process is based on the decomposition separation theorem where a lower-bound is added for the chance of going to one of the recurrent jets.

For an infinite word $w \in \mathbf{A}^\omega$ and $n \in \mathbb{N}$ we denote by $w_{<n}$ the prefix of length $n-1$ of w.

Definition 3.18 (Simple process).
We say that the process induced by $w \in \mathbf{A}^\omega$ from the state $s_0 \in \mathbf{S}$ is *simple* if there exists some $\mu > 0$ and decomposition of $\mathbf{S}^\omega$ into jets (A, B) such that,

- for all $n \in \mathbb{N}$ and $s \in A_n$, $\mathbb{P}(s_0 \xrightarrow{w_{<n}} s) \geq \mu$, and
- $\lim_{n\to\infty} \mathbb{P}(s_0 \xrightarrow{w_{<n}} B_n) = 0$.

An automaton for which for all words $w \in \mathbf{A}^\omega$ and states s_0, the induced process of w from s_0 is simple is called a *simple automaton*.

In [Chatterjee and Tracol, 2012] this structure is taken advantage of to show decidability of some of the problems. We are interested in how it compares to the leaks. First let us understand when a process is *not* simple.

Let us see an example of an automaton with a process that is not simple.

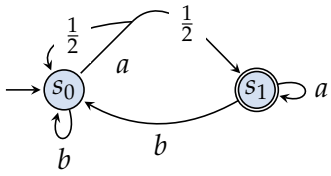


Figure 10: An automaton that is not simple

Example 12.
Even though the automaton in Figure 10 might look simple (it only has two states after all), *it is not*. We show how to induce a process that is not simple in this automaton. Define the finite words

$$w_n = a^n b.$$

Observe that $\mathbb{P}(s_0 \xrightarrow{a^n} s_0) = \frac{1}{2^n}$. On the other hand $\mathbb{P}(s_0 \xrightarrow{w_n} s_0) = 1$. This gives us a sequence with a sort of *pulsating* behavior. By reading a's, the chance to stay in s_0 decreases arbitrarily, then reading a b resets everything back to the initial state of the automaton and so on *ad infinitum*. The process induced by the infinite word

$$w = w_1 w_2 w_3 \cdots,$$

from the state s_0 is not simple. This is because the state s_0 has to appear infinitely often in at least one of the jets which renders impossible the simultaneous fulfillment of both conditions of a simple process.

Consequently the automaton in Figure 10 is not simple. △

Let $w \in \mathbf{A}^\omega$ and $s_0 \in \mathbf{S}$. One way that the process induced by w from s_0 is not simple is if there exists some state $s \in \mathbf{S}$ such that the probability to be in s from s_0 as we are progressively reading the word w, *pulsates*, just like in the example above. Let us illustrate what we mean by "pulsates". The sequence of the probabilities to go to state s from state s_0 in date n, that is $(\mathbb{P}(s_0 \xrightarrow{w_{<n}} s))_{n\in\mathbb{N}}$ goes to some $\epsilon_1 > 0$ then resets to some $\nu > \epsilon_1$, then goes back to ϵ_1, then to some $\epsilon_2 < \epsilon_1$, resets to ν and so on *ad infinitum*. The sequence of the probabilities to go from state s_0 back to s_0, in Example 12 with the word w is an instance of such a behavior. See the following figure.

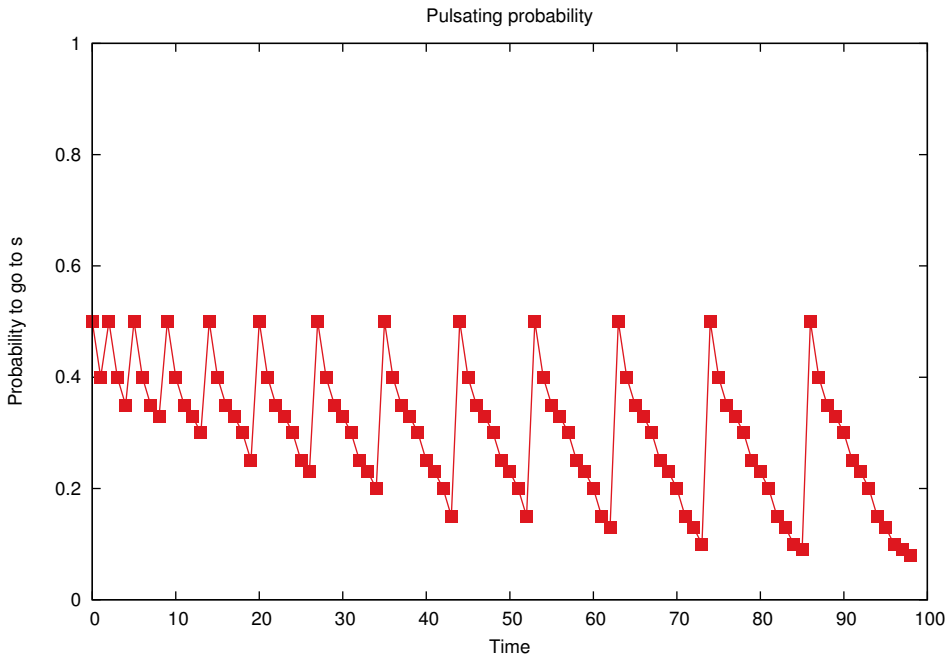


Figure 11: The probability to go to state s from state s_0 in a process that is not simple

One way to construct such a sequence is to take any sequence $(x_n)_{n\in\mathbb{N}}$ that converges to 0 and is decreasing and concatenate

$$x_1 \mid x_1, x_2 \mid x_1, x_2, x_3 \mid x_1, x_2, x_3, x_4 \cdots .$$

Why is such a process not simple? Assume that there exists some $\mu > 0$ and a decomposition of $\mathbf{S}^\omega$ into jets (A, B). Then for the process that we described above we see that there exists some $\nu > 0$ such that there are infinitely many $n \in \mathbb{N}$ for which

$$0 < \nu < \mathbb{P}(s_0 \xrightarrow{w_{<n}} s) < \mu. \tag{42}$$

Since (A, B) is a decomposition of $\mathbf{S}^\omega$ then for all the $n \in \mathbb{N}$ for which the inequation above holds, either $s \in A_n$ or $s \in B_n$, consequently there are two cases: either there are infinitely many $n \in \mathbb{N}$ for which (42) holds and $s \in B_n$, in which case the second item of Definition 3.18 cannot be true, or there exists one $n \in \mathbb{N}$ for which (42) holds and $s \in A_n$ then first item of Definition 3.18 does not hold. In any case such a process is not simple.

The automaton in Figure 10 is not simple, moreover, computing its extended Markov monoid one can conclude that it does not contain any leaks, and therefore it is leaktight. Thus we have:

Proposition 3.19.

$$LEAKTIGHT \not\subseteq SIMPLE.$$

Where by $LEAKTIGHT$ we denote the class of automata that are leaktight and by $SIMPLE$ the class of simple automata.

3.3 LEAKTIGHT AUTOMATA ARE A SUPERSET

In this section we will prove the main theorem of this chapter, that the class of simple automata is included in the class of leaktight automata.

Theorem 3.20.

$$SIMPLE \subset LEAKTIGHT.$$

Proof Idea.
The main point is that of the definition of a *non-simplicity witness*. This is a triple of elements of the Markov monoid with certain properties, that when they exist we can induce a process that has the pulsating behavior that we described for Example 12. In other words there is a sufficient condition that can be found in the Markov monoid (and computed) for the automaton to not be simple.

The proof of the theorem is in two steps. First we will prove that if the Markov monoid that is associated with an automaton has a non-simplicity witness then we can construct an infinite word that induces a process that is not simple. The second step is to show that when the Markov monoid contains a leak, then it also contains a non-simplicity witness, which, together with the first step, will conclude the proof of Theorem 3.20. △

We are going to need a theorem from [Fijalkow *et al.*, 2012], whose corollary is Theorem 3.14. The Markov monoid algorithm has no false positives, that is, when it replies *yes* then the value of the automaton is 1. Said in other words, the monoid is sound.

The elements of the monoid describe possible asymptotic behaviors of the automaton. For example, let $B \in \mathcal{M}$, and $s, s' \in \mathbf{S}$. If $B(s, s') = 1$, then there

exists a sequence of words such that asymptotically there is nonzero probability of going from s to s', and if $B(s, s') = 0$ then the chance of going from s to s' will tend to zero.

Definition 3.21 (Reification).
We say that the sequence of finite words $(w_n)_{n\in\mathbb{N}}$ *reifies* the element of the monoid $W \in \mathcal{M}$ if and only if for all $s, s' \in \mathbf{S}$, $(\mathbb{P}(s \xrightarrow{w_n} s'))_{n\in\mathbb{N}}$ converges and

$$W(s, s') = 1 \iff \lim_{n\to\infty} \mathbb{P}(s \xrightarrow{w_n} s') > 0.$$

The theorem of soundness of the Markov monoid algorithm can be stated as follows.

Theorem 3.22 ([Fijalkow *et al.*, 2012]).
For every element of the Markov monoid there exists a sequence of words that reifies it.

Theorem 3.14 follows as a corollary since there exists a sequence of words that reifies the value 1 witness. The *completeness* of the Markov monoid says the converse: if there exists a sequence of finite words that reaches a set of states, then it is accounted for in the monoid. This is not true in general, but it is true for leaktight automata.

The proof of Theorem 3.22 is by induction on the structure of the elements of the monoid. First it is easy to see that for all $a \in \mathbf{A}$, the elements B_a are reified by the constant sequences $(a)_{n\in\mathbb{N}}$. Then it is proved that the property remains true when taking products, *i.e.* $(w_n)_{n\in\mathbb{N}}$ reifies $W \in \mathcal{M}$ and $(z_n)_{n\in\mathbb{N}}$ reifies $Z \in \mathcal{M}$ then $(w_n z_n)_{n\in\mathbb{N}}$ reifies WZ. Finally if $(w_n)_{n\in\mathbb{N}}$ reifies the idempotent element $W \in \mathcal{M}$ then there exists an increasing function $f : \mathbb{N} \to \mathbb{N}$ such that $(w_{f(n)}^n)_{n\in\mathbb{N}}$ reifies $W^\#$. We will apply this theorem in the sequel, it is a way of going from the abstraction (elements of the monoid) to the actual behavior of the automaton (a sequence of finite words that reifies the element of the monoid).

In the classical result for Markov chains we partition the set of states into transient states, and recurrence classes. Transient states are states where the chance to go will diminish in the future, ultimately converging to zero. Recurrence classes are sets of states that are recurrent, and communicate with each

other, once the chain enters a recurrence class it stays there forever, and if it is aperiodic there will be a lower bound on the chance that the chain is in any one of the states of the recurrence class at every step. This bound is the gist of the following lemma, which we state and prove before moving on with the proof of Theorem 3.20.

Lemma 3.23.
Let $W \in \mathcal{M}$ be an idempotent element of the Markov monoid, $r \in \mathbf{S}$ that is W-recurrent and $(w_n)_{n\in\mathbb{N}}$ a sequence of words that reifies W. Then there exists a constant $\gamma > 0$ and a map that is strictly increasing $h : \mathbb{N} \to \mathbb{N}$ such that for all $n \in \mathbb{N}$,

$$\mathbb{P}(r \xrightarrow{w_{h(0)}\cdots w_{h(n-1)}} r) \geq \gamma.$$

Proof.
Let

$$\lambda = \frac{1}{2} \cdot \min\{\lim_n \mathbb{P}(s \xrightarrow{w_n} s') \mid W(s,s') = 1\}.$$

The constant λ is well-defined since by Definition 3.21 the limits exist for all $s, s' \in \mathbf{S}$. Observe that since $(w_n)_{n\in\mathbb{N}}$ reifies W there exists an increasing function $h : \mathbb{N} \to \mathbb{N}$ such that for all $n \in \mathbb{N}$, and $s, s' \in \mathbb{N}$

$$W(s,s') = 0 \implies \mathbb{P}(s \xrightarrow{w_{h(n)}} s') < \frac{1}{|\mathbf{S}| \cdot 2^{n+2}}, \text{ and} \tag{43}$$

$$W(s,s') = 1 \implies \mathbb{P}(s \xrightarrow{w_{h(n)}} s') \geq \lambda. \tag{44}$$

Set $\gamma = \frac{\lambda}{2}$ and $z_n = u_{h(0)}u_{h(0)} \cdots u_{h(n-1)}$. Denote by R the recurrence class of r *i.e.*

$$R = \{t \in \mathbf{S} \mid W(r,t) = 1\}.$$

Since W is idempotent the set R is closed, that is to say that for every $s \in R$ and $s' \in \mathbf{S}$, $W(s,s') = 1$ implies that $s' \in R$. Consequently using (43) we can show that

$$\mathbb{P}(r \xrightarrow{z_{n-1}} \mathbf{S} \setminus R) < \frac{1}{2}.$$

This follows because

$$\mathbb{P}(r \xrightarrow{z_{n-1}} \mathbf{S} \setminus R) \leq \sum_{s\in R} \sum_{k=0}^{n-2} \mathbb{P}(s \xrightarrow{w_{h(k)}} \mathbf{S} \setminus R) < \sum_{s\in R} \sum_{k=0}^{n-2} \frac{1}{|\mathbf{S}| \cdot 2^{k+2}} < \frac{1}{2}.$$

Hence

$$\mathbb{P}(r \xrightarrow{z_{n-1}} R) \geq \frac{1}{2}.$$

Since r is W-recurrent and W is idempotent for all $s \in R$ we have $W(s,r) = 1$ consequently using (44) we conclude that

$$\mathbb{P}(r \xrightarrow{z_n} r) \geq \sum_{s \in R} \mathbb{P}(r \xrightarrow{z_{n-1}} s)\mathbb{P}(s \xrightarrow{w_{n-1}} r) \geq \frac{1}{2} \cdot \lambda.$$

■

We define the central notion, that of a *non-simplicity witness*.

Definition 3.24 (Non-simplicity witness).
A tripe (U, V, W) of elements of the Markov monoid $\mathcal{M}$ is a *non-simplicity witness* if there exist states $r, t \in \mathbf{S}$ such that

- $UV^{\#}W$ is idempotent,
- r is $UV^{\#}W$-recurrent,
- $UV(r,t) = 1$,
- t is V-transient.

Now we move on with the proof of Theorem 3.20 in two steps, first, if the Markov monoid associated to an automaton has a non-simplicity witness then the automaton is not simple, second, if the Markov monoid has a leak then it also has a non-simplicity witness.

In order to prove the first step, we make the idea of pulsating sequence discussed in the previous section more formal using the lemma that follows. For an infinite word $w = a_0a_1a_2\cdots$ we use this notation of subwords, for $j, k \in \mathbb{N}$, $j < k$

$$w[j,k] = a_j \cdots a_{k-1}.$$

Lemma 3.25.
Let $w \in \mathbf{A}^{\omega}$ be an infinite word. If there exists states $p, s, t \in \mathbf{S}$, increasing sequences $(i_n)_{n\in\mathbb{N}}$, $(j_n)_{n\in\mathbb{N}}$ and $\gamma > 0$ such that

1. for all $n \in \mathbb{N}$, $\mathbb{P}(p \xrightarrow{w_{<i_n}} s) \geq \gamma$,
2. for all $n \in \mathbb{N}$, $i_n < j_n$ and $\mathbb{P}(s \xrightarrow{w[i_n,j_n]} t) > 0$,
3. $\lim_{n\to\infty} \mathbb{P}(p \xrightarrow{w_{<j_n}} t) = 0$,

then the process induced by w from p is not simple.

Proof.
We assume on the contrary that w induces a simple process from p with bound λ and show a contradiction.

Observe first that for infinitely many $n \in \mathbb{N}$, we have $s \in A_{i_n}$ and $t \in B_{j_n}$, where A and B are the jets from the Definition 3.18. This follows from the assumptions (1) and (3).

Let $n \in \mathbb{N}$ be such that $s \in A_{i_n}$ and $t \in B_{j_n}$, (2) implies that

$$\mathbb{P}(s \xrightarrow{w[i_n, j_n]} t) > 0,$$

therefore along the path from s to t there must be some moment where we move from the jet A to the jet B. Formally there exists some k_n such that $i_n \leq k_n < j_n$, $s_{k_n} \in A_{k_n}$, $s_{k_n+1} \in B_{k_n+1}$ and there is nonzero probability to go from s_{k_n} to s_{k_n+1} with the letter $w[k_n, k_n + 1] = b$. Denote by $p_{\min}$ the smallest nonzero transition probability in the automaton, that is

$$p_{\min} = \min\{\delta(s, a)(s') > 0 \mid s, s' \in \mathbf{S}, a \in \mathbf{A}\}.$$

Then

$$\mathbb{P}(s_{k_n} \xrightarrow{b} s_{k_n+1}) \geq p_{\min}.$$

Now because $s_{k_n} \in A_{k_n}$,

$$\mathbb{P}(p \xrightarrow{w_{<k_n+1}} s_{k_n+1}) \geq \mathbb{P}(p \xrightarrow{w_{<k_n}} s_{k_n}) \cdot \mathbb{P}(s_{k_n} \xrightarrow{b} s_{k_n+1}) \geq \lambda \cdot p_{\min}.$$

Since this holds for infinitely many $n \in \mathbb{N}$ and $s_{k_n+1} \in B_{k_n+1}$ this contradicts the second point in Definition 3.18, that of $\lim_{n\to\infty} \mathbb{P}(p \xrightarrow{w_{<n}} B_n) = 0$. ∎

Now we show that when the Markov monoid has a non-simplicity witness we can construct an infinite word such that the conditions in Lemma 3.25 are fulfilled.

Lemma 3.26.
If the Markov monoid associated to an automaton has a non-simplicity witness then it is not a simple automaton.

Proof.
Let $(U, V, W) \in \mathcal{M}$ be a non-simplicity witness with states $r, t \in \mathbf{S}$ for which the properties in Definition 3.24 are true. From Theorem 3.22 let $(u_n)_{n\in\mathbb{N}}$, $(v_n)_{n\in\mathbb{N}}$, $(w_n)_{n\in\mathbb{N}}$ be sequences of words that reify U, V and W respectively. As mentioned in the discussion of the proof of Theorem 3.22, there exists an

increasing function $f : \mathbb{N} \to \mathbb{N}$ such that $(v^n_{f(n)})_{n\in\mathbb{N}}$ reifies $V^\#$. The proof of this, can be found in [Fijalkow *et al.*, 2012, Lemma 4].

Then $(v_{f(n)})_{n\in\mathbb{N}}$ reifies V as well, since it is a subsequence of $(v_n)_{n\in\mathbb{N}}$.

If $(x_n)_{n\in\mathbb{N}}$ reifies X and $(y_n)_{n\in\mathbb{N}}$ reifies Y, it is easy to prove that $(x_n y_n)_{n\in\mathbb{N}}$ reifies XY. Consequently since V is idempotent, for all $k \in \mathbb{N}$, $(v^k_{f(n)})_{n\in\mathbb{N}}$ reifies $V^k = V$. By assumption $UV(r,t) = 1$ so there exists $N_k \in \mathbb{N}$ such that for all $n \geq N_k$,

$$\mathbb{P}(r \xrightarrow{u_n v^k_{f(n)}} t) > 0.$$

Let $g(n) = \max(n, N_n)$. We omit the parentheses or function composition operator and write $fg(n)$ for $f(g(n))$. Since g is increasing, $(u_{g(n)} v^{g(n)}_{fg(n)} w_{g(n)})_{n\in\mathbb{N}}$ is a subsequence of $(u_n v^n_{f(n)} w_n)_{n\in\mathbb{N}}$ so it reifies $UV^\#W$ as well. We will use this sequence of finite words to construct an infinite word that induces a process that is not simple. Observe that by the definition of g,

$$\text{for all } n \in \mathbb{N}, \mathbb{P}(r \xrightarrow{u_{g(n)} v^n_{fg(n)}} t) > 0. \tag{45}$$

We apply Lemma 3.23 to the element $UV^\#W$, the state r and the sequence of words $(u_{g(n)} v^{g(n)}_{fg(n)} w_{g(n)})_{n\in\mathbb{N}}$ to obtain the bound $\gamma > 0$ and h another increasing function. Denote the following shorthand,

$$(z_n)_{n\in\mathbb{N}} = (u_{gh(n)} v^{gh(n)}_{fgh(n)} w_{gh(n)})_{n\in\mathbb{N}},$$
$$(x_n)_{n\in\mathbb{N}} = (u_{gh(n)} v^{gh(n)}_{fgh(n)})_{n\in\mathbb{N}}.$$

We have

$$\text{for all } n \in \mathbb{N}, \mathbb{P}(r \xrightarrow{z_0 \cdots z_{n-1}} r) \geq \gamma. \tag{46}$$

We argue that the conditions of Lemma 3.25 are met:

1. for all $n \in \mathbb{N}$, $\mathbb{P}(r \xrightarrow{z_0 \cdots z_{n-1}} r) \geq \gamma$,
2. for all $n \in \mathbb{N}$, $\mathbb{P}(r \xrightarrow{x_n} t) > 0$, and
3. $\lim_{n\to\infty} \mathbb{P}(r \xrightarrow{z_0 \cdots z_{n-1} x_n} t) = 0$.

Items 1 and 2 follow from (46) and (45) respectively. We prove item 3.

Observe that $(x_n)_{n\in\mathbb{N}}$ reifies $UV^\#$, since we are only taking subsequences of the original sequence. Let $s \in \mathbf{S}$, since t is V-transient from the hypothesis, $UV^\#(s,t) = 0$, and because $(x_n)_{n\in\mathbb{N}}$ reifies $UV^\#$, $\lim_{n\to\infty} \mathbb{P}(s \xrightarrow{x_n} t) = 0$. We conclude the proof of item 3 because

$$\mathbb{P}(r \xrightarrow{z_0 \cdots z_{n-1} x_n} t) = \sum_{s\in\mathbf{S}} \mathbb{P}(r \xrightarrow{z_0 \cdots z_{n-1}} s) \cdot \mathbb{P}(s \xrightarrow{x_n} t),$$

and the second factor of every term is tending to zero.

So we can apply Lemma 3.25 for the infinite word $z = z_0 z_1 \cdots$, and from it we deduce that the process induced by z from r is not simple. ∎

Now we proceed to the second and last part of the proof of Theorem 3.20. We will show that if the Markov monoid has a leak then it has a non-simplicity witness.

For the proof we need the notion of *#-height*. For some subset of the monoid $R \subseteq \mathcal{M}$ denote by $[R]$ the smallest set that contains all elements of R and is closed under taking product. Let $E(R)$ be the set of idempotent elements in R.

Definition 3.27 (#-height, [Fijalkow *et al.*, 2015]).
Set

$$S_0 = [\{B_a \mid a \in \mathbf{A}\}]$$
$$S_n = [S_{n-1} \cup \{W^{\#} \mid W \in E(R)\}].$$

The #-height of some $W \in \mathcal{M}$ is the smallest n for which $W \in S_n$.

Lemma 3.28.
If the extended Markov monoid of an automaton $\widetilde{\mathcal{M}}$ contains a leak then it also contains a non-simplicity witness.

Proof.
Let $\mathcal{L} \subseteq \widetilde{\mathcal{M}}$ be the set of all elements $(Z, \widetilde{Z})$ for which there exists $r, r' \in \mathbf{S}$ such that

1. $(Z, \widetilde{Z})$ is idempotent,
2. r is Z-recurrent,
3. $Z(r, r') = 0$,
4. $\widetilde{Z}(r, r') = 1$.

By hypothesis $\mathcal{L}$ is not empty because we have assumed that $\widetilde{\mathcal{M}}$ contains a leak, and all leaks are contained within $\mathcal{L}$. Let $(Z, \widetilde{Z}) \in \mathcal{L}$ such that no other element of $\mathcal{L}$ has a strictly smaller #-height, and $r, r' \in \mathbf{S}$ such that the enumerated items above hold.

Since $Z \neq \widetilde{Z}$, there must exist $U, V, W \in \mathcal{M}$ such that $Z = UV^{\#}W$ and $V \neq V^{\#}$. Let T be the set of states that are V-transient and $R = \{s \in \mathbf{S} \mid Z(r,s) = 1\}$, the Z-recurrence class of r. We will prove that

$$\text{there exists } r' \in R \text{ and } t' \in T \text{ such that } UV(r',t') = 1. \tag{47}$$

Observe that this is sufficient for proving that (U,V,W) is a non-simplicity witness, the states r' and t' fulfill the conditions in Definition 3.24.

We prove (47) by contradiction. Assume that it does not hold, we will show that the element of the extended Markov monoid $(UVW, \widetilde{Z})^{|\widetilde{\mathcal{M}}|!}$ is in $\mathcal{L}$. This is a contradiction because UVW has a strictly smaller #-height than $Z = UV^{\#}W$.

So what is left to show is that $(UVW, \widetilde{Z})^K$ is in $\mathcal{L}$, where $K = |\widetilde{\mathcal{M}}|!$.

Observe first that from the assumption that (47) is not true, for all $r' \in R$ and $s \in \mathbf{S}$, $UV(r',s) = UV^{\#}(r',s)$ and from here

$$UVW(r',s) = UV^{\#}W(r',s) = Z(r',s). \tag{48}$$

We demonstrate that $(UVW, \widetilde{Z})^K \in \mathcal{L}$, for the states r, r'.

1. $(UVW, \widetilde{Z})^K$ is idempotent:

   This follows from the fact that for every finite monoid of size n and element x, $x^{n!}$ is idempotent.

2. r is $(UVW)^K$-recurrent:

   Let $s \in \mathbf{S}$ such that $(UVW)^K(r,s) = 1$. From idempotency, $r \in R$, therefore applying (48), $(UV^{\#}W)^K(r,s) = 1$. From here, since $(UV^{\#}W)^K = Z^K = Z$, $Z(r,s) = 1$. Because r is $Z-$*recurrent*, $Z(s,r) = 1$, and $s \in R$. Applying (48) again, this time for s, we have $Z(s,r) = UVW(s,r) = 1$. Also $1 = Z(r,r) = UVW(r,r)$, hence $(UVW)^K(s,r) = 1$.

3. $(UVW)^K(r,r') = 0$:

   Assume on the contrary that $(UVW)^K(r,r') = 1$. Then there exist $s_0 \cdots s_K$ such that $s_0 = r, s_K = r'$ and for all $0 \leq i < k$, $UVW(s_i, s_{i+1}) = 1$. Applying (48) for $i = 0$, since $s_0 = r \in R$ we see that $s_1 \in R$ as well, because r is Z-recurrent. Inductively we conclude that for all $0 \leq i < k$, $Z(s_i, s_{i+1}) = 1$, hence $Z^K(r,r') = 1$, Z is idempotent thus $Z(r,r') = 1$, a contradiction of point (3) in the definition of $\mathcal{L}$.

4. $\widetilde{Z}^K(r,r') = 1$:

   This follows immediately from point (4) of the definition of $\mathcal{L}$ and the fact that $\widetilde{Z}$ is idempotent. ∎

In this way we conclude the demonstration of Theorem 3.20. The class of leaktight automata is also closed under synchronized product, parallel composition etc, [Fijalkow *et al.*, 2015]. Since there are no known classes of automata with decidable value 1 problem that are not a strict subset of leaktight and since the Markov monoid algorithm that takes advantage of this class is in some way the best that we can hope for [Fijalkow, 2015] we can safely conclude that this class is robust and deserves attention. It seems to capture tightly the complications that stem from calculating infinitely precise transition probabilities, that are the cause of undecidability of the value 1 problem. In the next chapter we will continue with the notion of leaks and lift it to two player half-blind games, where it proves just as useful as for probabilistic automata.

3.4 STAMINA, THE TOOL

Stamina (stabilization monoids in automata theory) [Fijalkow *et al.*, 2016] is a tool authored by Nathanaël Fijalkow, Hugo Gimbert, Denis Kuperberg and the author of this thesis that computes stabilization monoids. An example of a stabilization monoid is the Markov monoid that we have discussed in this chapter. Stamina is a successor to the tool ACME [Fijalkow and Kuperberg, 2014]. The aim of Stamina is to solve the following three problems:

- The value 1 problem for leaktight probabilistic automata. It implements the Markov monoid algorithm, that computes the Markov monoid. The size of the Markov monoid can be exponential in number of states. As a consequence, having a small memory footprint is crucial. Stamina achieves this by a set of techniques, such as saving the same matrix row only once in the memory, and managing only pointers to the locations of the saved row. This enables us to handle relatively large automata despite the $PSPACE$ complexity of deciding the value 1 problem for leaktight automata.

- The star height problem. We are given a finite automaton and an integer, we have to decide whether the language defined by the automaton has star height equal to the integer. The star height of a language is the smallest number of nested Kleene stars that are necessary for expressing the language as a regular expression without complements. The decidability of the star height problem was an open question for 25 years until Hashiguchi answered it positively in [Hashiguchi, 1988]. Though he showed that the problem is decidable, his algorithm has a computational complexity that makes implementing it worthless. After an equally long time, Kirsten gave a simpler algorithm with better complexity bounds in [Kirsten, 2004], using a stabilization monoid. Even though this algorithm

is much simpler than the first one by Hashiguchi, it remains far from tractable being doubly exponential in space. In order to ameliorate the situation we employ the *loop complexity* heuristic. Overall the star height problem remains computationally hard but Stamina provides a sandbox for experimenting with different heuristics and verifying small examples.

- Boundedness problem for regular cost functions [Colcombet, 2009]. We are given an automaton with counters that defines some function $f : \mathbf{A}^* \to \mathbb{N}$, where $\mathbf{A}$ is the alphabet of the automaton, and asked to determine if f is bounded. This problem reduces to computing a stabilization monoid as well.

The three problems: the value 1 problem for probabilistic automata, the star height problem, and the boundedness problem for regular cost functions have been glued together in the sense that we have one tool to solve the three of them, because the three problems reduce to computing stabilization monoids. The algorithms that solve them are essentially the same.

Stamina can be run as a standalone application (it can output PDFs of the stabilization monoids it computes), as well as a module for SAGE (`http://www.sagemath.org`). In this way it can be used in conjunction with the whole platform of SAGE in general, and its automata libraries in particular.

Stamina furthers the work that was done for ACME, both in terms of features and problems that are solved, as well as, perhaps more importantly performance. This is evident by the benchmarks that were performed and then plotted in Figure 12. In particular, we see that there is a threshold around the point of 3000 element monoids, which cannot be passed by ACME because of stack overflows.

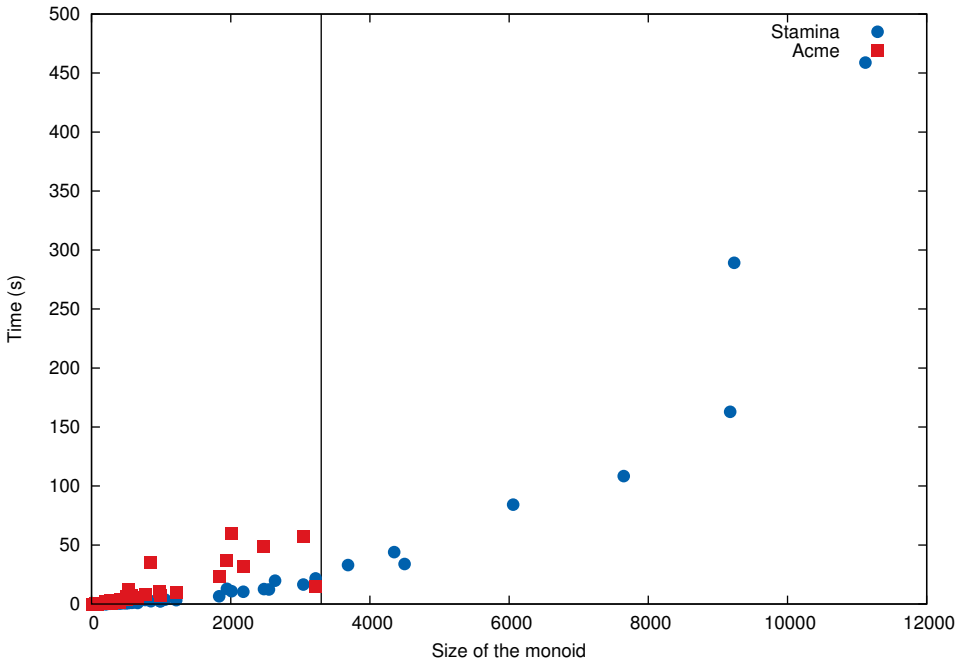


Figure 12: Benchmark comparison between Stamina and ACME

4

HALF-BLIND LEAKTIGHT GAMES

Half-blind stochastic games are finite duration games that are played on a finite arena between two players. The maximizer that wants to reach the final set of states and the minimizer that wants to do the opposite. They take turns for a finite but arbitrarily long duration. The difference between simple stochastic games and half-blind ones is that in half-blind games we restrict the maximizer to a particular set of strategies. The maximizer can choose a pure and finite word over the alphabet of the actions as his strategy. Whereas his opponent chooses among general behavioral strategies. In other words we are dealing with a special case of two player games with partial information. The case where the maximizer has zero information (he is blind), while the minimizer is perfectly informed. From another point of view, half-blind games are generalizations of probabilistic automata on finite words. They are probabilistic automata with an adversary that tries to stop the protagonist from reaching the set of final states. The problem that we are going to study for these games in this chapter is the *maxmin reachability problem*. This is the question of whether for all $\epsilon > 0$ there exists a finite word for maximizer such that against all strategies of the minimizer, the chance of reaching the set of final states is larger than $1 - \epsilon$. The maxmin reachability problem is the analogue of the value 1 problem for probabilistic finite automata. Consequently it is not decidable in general. Nevertheless, we will identify the subclass of *leaktight half-blind games* for which the maxmin reachability problem is decidable. This lifts the result of [Fijalkow *et al.*, 2012] to two player half-blind games. We will also compare the power of different types of strategies and show that mixed strategies are in general stronger for both players, as well as the perhaps surprising fact that optimal strategies of the minimizer might require infinite memory even though he plays against an opponent that has zero information.

4.1 OVERVIEW

Two player stochastic games are suitable for modeling numerous phenomena with probability. In the field of software verification and controller synthesis, two player stochastic games with *partial information* are particularly useful. The game is played between the controller and the environment. By taking actions the former tries to reach his objective (put the system in some state, see a set of states infinitely often, etc.), and the environment tries to stop this. The controller takes actions that are helpful to reaching his objective based on the history of states. Albeit, typically in practice the controller cannot know the exact state of the system, but has only a partial picture. This is because the controller is aided by sensors, which can be noisy, or have some degree of error; or a software interface that provides only a partial view of the system. Hence the suitability of partial information two player stochastic games in this very important context.

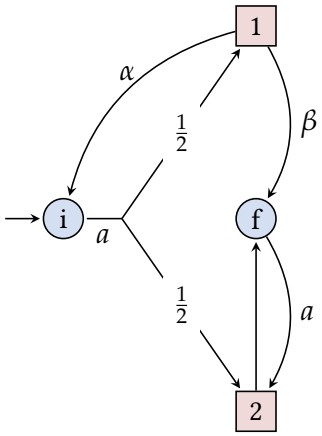


Figure 13: A half-blind game

In this chapter we will discuss a particular type of two player partial information stochastic game: the half-blind games. One arrives to these games by generalizing probabilistic finite automata to two player games, with a perfectly informed opponent.

Consider Figure 13. This arena looks like the arenas of two player games that we have discussed in Chapter 2 — we see a turn-based stochastic game played between two players on a finite graph. One player's objective (Max) is to reach the final state f, his opponent (Min) tries to stop him from doing this. As in probabilistic finite automata, Max has no information about the state of the game, and his aim is to reach the set of final states by playing some *finite* word over the set of actions. His opponent, Min, on the other hand is perfectly

informed about the state of the game. He can see the whole history of the states and actions that were taken before he makes the decision, when it is his turn.

The question that we will study is that of whether the set of final states is *maxmin reachable* from the initial state. This is analogous to the value 1 problem for probabilistic finite automata. Before we define this problem precisely, let us overview related work.

Partial information games are algorithmically harder than their perfect information counterparts; especially quantitative problems, they are often undecidable. We saw in the previous chapter how many problems are undecidable for probabilistic automata, which is a special case of a partial information game with one player. But there is a series of positive results for qualitative problems.

For example, Büchi partial information games are qualitatively determined [Bertrand *et al.*, 2009], *i.e.* either Max can ensure the Büchi condition almost surely or Min can ensure positively the opposite co-Büchi condition. In a reachability game, there are three possibilities, either Max can ensure the condition almost surely, Min can ensure the opposite safety condition surely, or both players have positively winning strategies. Moreover for both games, only finite-memory strategies are necessary, and which player has an almost-sure or positively winning strategy can be decided, and the strategy that ensures it, can be effectively constructed as well.

Consult the survey article [Chatterjee *et al.*, 2012] and the references therein, for a general overview of two player games with partial information. It suffices to illustrate the contrast between the perfect information and partial information games: for example the problem of the existence of a strategy that ensures almost sure reachability (of a set of states) for perfect information games is in linear time, whereas for partial information games it is $2EXPTIME$-complete. For perfect information games with parity condition, the question whether there exists a strategy that ensures the parity condition almost surely is in $NP \cap coNP$, but for partial information games it is undecidable.

There is the interesting special case of *one-sided games*. Partial information games where the minimizer is assumed to have perfect information, and the maximizer has partial information. In the context of software verification these are games where the antagonistic environment is perfectly informed, a pessimistic but safe approach. Under this restriction there is some relief. Deterministic two player one-sided games with parity objectives are in $NP \cap coNP$, whereas the general case of deterministic partial observation games is $2EXPTIME$-complete [Chatterjee and Henzinger, 2005].

In the stochastic case, even the one-sided games for parity objectives under the almost-sure semantics are undecidable. However, the difficulty stems from the analysis of strategies with infinite memory. In many applications such strategies are useless, since we cannot actually implement them. Whether the protagonist can satisfy the parity condition almost surely with a finite-memory

strategy in a one-sided stochastic game, was proved to be decidable in [Nain and Vardi, 2013]. This problem is $EXPTIME$-complete and the strategies need to have memory of at most exponential size [Chatterjee *et al.*, 2014]. We are going to take the same approach and assume that the opponent (Min) is perfectly informed.

In the majority of the positive results, the almost-sure semantics is being considered, *i.e.* does there exist a strategy for Max such that against all strategies for Min the winning condition is ensured with probability 1. One can argue that this question is at times too strong, in comparison to the limit-sure semantics. Especially for one-sided games, where we already assume that Min is perfectly informed. Moreover, there are simple examples of games with reachability condition where there is no strategy that ensures that the set of final states will be reached almost surely, but for every $\epsilon > 0$ there is some strategy that ensures it with probability of at least $1 - \epsilon$. One such game is depicted in Figure 13.

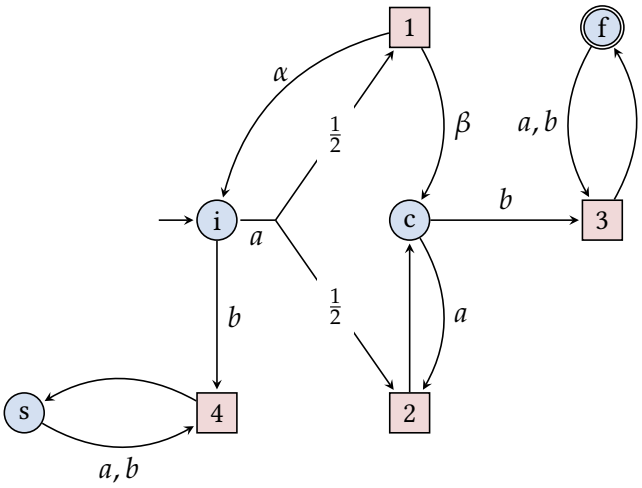


Figure 14: A half-blind game with no strategy for Max that wins almost surely

There is no strategy for Max that ensures reaching the state f almost surely. Because Min can always play the action α, so whenever Max plays a b, the game ends up with some nonzero probability in the sink state s. On the other hand the state f is limit-sure reachable: Max plays a sequence of n a's which results in the game being in state c with probability at least $1 - 2^{-n}$, and then plays a b. So f is limit-sure reachable but it is not almost sure reachable.

The reason for lack of treatment of the limit-sure semantics in the literature is that even in the case of probabilistic finite automata the limit-sure problem (value 1 problem) is undecidable. However, we saw in the previous chapter that there are at least a couple of interesting classes of probabilistic automata with

decidable value 1 problem. It is interesting to see whether we can lift these results to games.

In this chapter we will use the notion of a leak to show decidability of the maxmin reachability problem for a class of half-blind games. Precisely, the maxmin reachability problem is as follows.

Problem 4.1 (Maxmin reachability for half-blind games).
Given a half-blind game, decide whether for all $\epsilon > 0$ there exists a finite word (over the alphabet of actions) for Max, such that against any strategy for Min, the set of final states is reached with probability of at least $1 - \epsilon$.

Leaks seem to catch well the difficulties that make the value 1 problem undecidable for probabilistic finite automata. It is interesting to explore whether in games it is this phenomena of leaks that alone causes undecidability, or if there is more hardness added with the perfectly informed opponent. Another motivation for considering half-blind games is as follows.

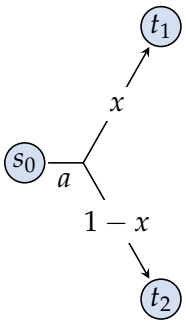


Figure 15: An uncertain transition in a PFA

With half-blind games we can model lack of knowledge of exact transition probabilities. Imagine that in the transition in Figure 15 we do not know exactly the value of x, but we know that it is in some 2ϵ neighborhood of $\frac{1}{2}$.

We model this lack of knowledge by basically letting Min make the choice, as in Figure 16.

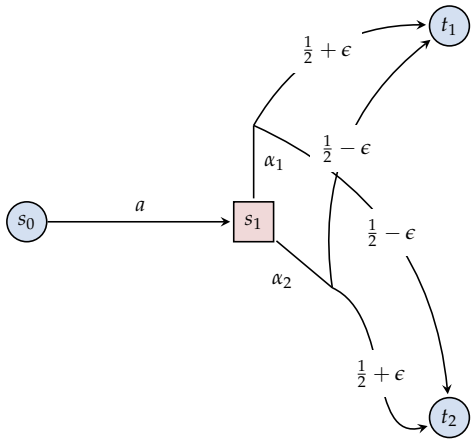


Figure 16: Translation to half-blind games

In the sequel we will report on [Kelmendi and Gimbert, 2016], identify the class of leaktight half-blind games, then proceed to prove that the maxmin reachability problem in this restricted subclass is decidable. Moreover we will compare the power of different types of strategies and prove that optimal strategies for the minimizer might require infinite memory.

The proof methods that we will employ are as follows. We will define the *belief monoid*, which is a finite monoid that we will build on top of the Markov monoid that was used for probabilistic finite automata. It will play the same role that the Markov monoid plays for probabilistic automata, namely we will search it for a particular element whose existence is equivalent to the maxmin reachability of the set of final states. In order to demonstrate that the belief monoid is sound, we utilize a modification of the Simon's factorization forest theorem [Simon, 1990], for a data structure that is called the k-factorization tree. With the help of k-factorization trees we will prove some upper and lower bounds of probabilities in the game. This is however done under the assumption that the game is leaktight, *i.e.* it is not guaranteed that there are no false positives for the general case of half-blind games. This stands in contrast to [Fijalkow *et al.*, 2012], where the soundness of the Markov monoid is almost for free, and does not rely on the leaktight hypothesis. Consequently we will prove that the belief monoid is complete, which uses the leaktight hypothesis and Simon's forest factorization theorem as well.

4.2 THE BELIEF MONOID

4.2.1 *Definitions and values*

The arena of a half-blind game is very similar to that of perfect information games. We will however assume that the game is played in a bipartite graph, in order to make the presentation cleaner.

We define the half-blind games formally.

Definition 4.2 (Half-blind game).
A half-blind game $\mathcal{G}$ is given by the tuple $\mathcal{G} = (\mathbf{S_1}, \mathbf{S_2}, \mathbf{A}_1, \mathbf{A}_2, p, \mathbf{F})$, where

- $\mathbf{S_1}$ is a finite set of states that is controlled by Max,
- $\mathbf{S_2}$ is a finite set of states that is controlled by Min,
- $\mathbf{A}_1$ is a finite set of actions that are disposable to Max,
- $\mathbf{A}_2$ is a finite set of actions that are disposable to Min,
- p is the transition function that maps $\mathbf{S_1} \times \mathbf{A}_1$ to $\Delta(\mathbf{S_2})$, and $\mathbf{S_2} \times \mathbf{A}_2$ to $\Delta(\mathbf{S_1})$,
- $\mathbf{F}$ is a set of final states.

We will write $\mathbf{S} = \mathbf{S_1} \cup \mathbf{S_2}$ and $\mathbf{A} = \mathbf{A}_1 \cup \mathbf{A}_2$. The objective of Max is to reach the set of final states $\mathbf{F}$.

The game proceeds in turns. Say that it starts in some state s_0, if $s_0 \in \mathbf{S_1}$ then Max chooses some action $a \in \mathbf{A}_1$ and the next state is determined from the distribution $p(s_0, a)$, symmetrically if $s_0 \in \mathbf{S_2}$. Then Min chooses his action and so on until Max decides to stop, and when he does, if the current state of the game is in $\mathbf{F}$ then he wins, otherwise it is Min who wins. The decision of what action to take, Min can base on the whole history, including the actions that he himself took, as well as those of Max. But Max can base his own decisions only on the number of turns that have elapsed.

In other words, Max plays a finite and pure word over the alphabet $\mathbf{A}_1$. Min's strategy is a general behavior one.

Definition 4.3 (Strategies for half-blind games).
The set of strategies for Max is denoted by Σ_1 and it is defined as

$$\Sigma_1 = \mathbf{A}_1^*,$$

the set of pure finite words over the alphabet $\mathbf{A}_1$.

The set of strategies for Min is denoted by Σ_2 and it is the set of functions

$$(\mathbf{SA})^*\mathbf{S_2} \to \Delta(\mathbf{A}_2).$$

We will usually denote by w the elements of Σ_1, to stress that it is a particular kind of strategy, *i.e.* a word, and by τ the elements of Σ_2. Why such strategies have been chosen, and discussions of other types of strategies can be found in Section 4.7.

When we fix an initial state $s \in \mathbf{S}$, a strategy τ for Min and a finite word $w \in \mathbf{A}_1^*$ of length n for Max, we induce a probability distribution on the set $(\mathbf{SA})^n\mathbf{S_1}$. We will denote this by $\mathbb{P}_s^{w,\tau}$. More precisely it is defined as follows: for a history $h = s_1a_1t_1b_1 \cdots s_na_nt_nb_ns_{n+1} \in (\mathbf{SA})^n\mathbf{S_1}$,

$$\mathbb{P}_s^{w,\tau}(h) = \prod_{i=1}^{n} p(s_i, a_i)(t_i) \cdot \tau(h_i)(b_i) \cdot p(t_i, b_i)(s_{i+1}),$$

if $s = s_1$ and $w = a_1 \cdots a_n$ and 0 otherwise, where $h_i = s_1a_1t_1b_1 \cdots s_ia_it_i$, $1 \leq i \leq n$.

As we did before, we will use the following shorthand. For $t \in \mathbf{S_1}$ we denote by $\mathbb{P}_s^{w,\tau}(t)$ the chance of being in state t, when the players have chosen the respective strategies w and τ, and the game starts from the state s, *i.e.*

$$\mathbb{P}_s^{w,\tau}(t) = \sum_{h \in (\mathbf{SA})^*} \mathbb{P}_s^{w,\tau}(ht).$$

For a set $T \subseteq \mathbf{S}$, we write $\mathbb{P}_s^{w,\tau}(T)$ for the chance of ending up in the set of states T, *i.e.*

$$\mathbb{P}_s^{w,\tau}(T) = \sum_{t \in T} \mathbb{P}_s^{w,\tau}(t).$$

We are interested in the *maxmin value* of the game. This is defined as follows.

Definition 4.4 (Maxmin value).
Let $\mathcal{G}$ be a game, with $\mathbf{F}$ its set of final states and $s \in \mathbf{S}$ some state. We define the maxmin value of s to be

$$\underline{val}(s) = \sup_{w \in \Sigma_1} \inf_{\tau \in \Sigma_2} \mathbb{P}_s^{w,\tau}(\mathbf{F}).$$

The problem that we will consider is whether $\underline{val}(s) = 1$, where s is the initial state. This is Problem 4.1, that is the problem of whether for all $\epsilon > 0$ there exists some word w such that against all strategies τ of the opponent the set of final states $\mathbf{F}$ is reached with probability of at least $1 - \epsilon$. So player Min knows the strategy that was chosen by Max before making his own strategy choice. In the controller synthesis application, this results in a controller that wins even against a strong environment, which is desirabl.e

In general $\sup_{w \in \sigma_1} \inf_{\tau \in \Sigma_2} \mathbb{P}_s^{w,\tau}(\mathbf{F})$ is not equal to $\inf_{\tau \in \Sigma_2} \sup_{w \in \sigma_1} \mathbb{P}_s^{w,\tau}(\mathbf{F})$, it matters who chooses the strategy first. But if instead of considering only pure words for Max, we consider mixed words as well, that is elements of $\Delta(\Sigma_1^*)$, then it does not matter who chooses the strategy first, *i.e.* the game has a value. We will discuss this choice, as well as how much memory the strategies require in Section 4.7. For now we concentrate only on the maxmin value.

Unfortunately it is not possible to algorithmically determine if a game has maxmin value equal to 1.

Theorem 4.5 ([Gimbert and Oualhadj, 2010]).
Given a half-blind game $\mathcal{G}$, and $s \in \mathbf{S}$, the problem of whether

$$\underline{val}(s) = 1,$$

is undecidable.

Observe that if Min has no choice in any of the states $\mathbf{S_2}$ (there is a single action), then we are dealing with the special case of probabilistic finite automata. The value 1 problem is undecidable, hence the theorem above. Our purpose is to decide whether $\underline{val}(s) = 1$ when the game is leaktight.

4.2.2 *An Example*

Let us explore a more involved example.

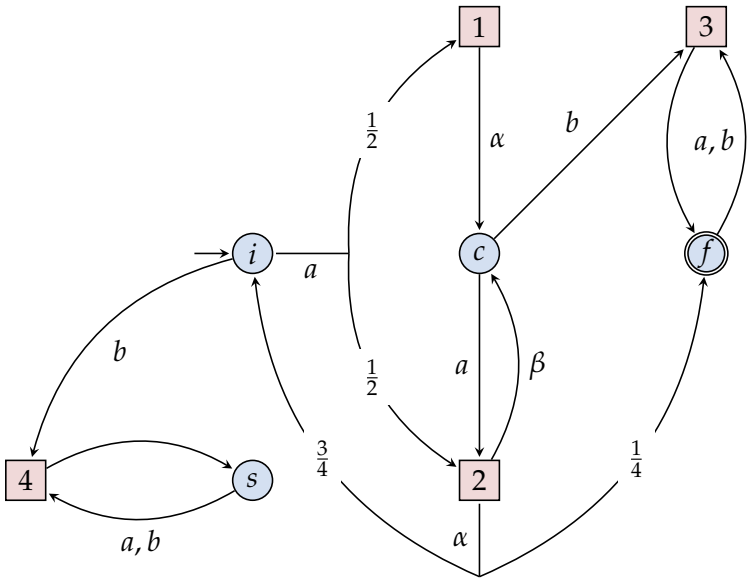


Figure 17: A half-blind game with $\underline{val}(i) < 1$

Example 13.
The game starts at state i and the objective of the maximizer is to reach the state f, which is the unique final state. Observe that in order to reach the state f, Max has to play the letter b at some point, because against the strategy that plays only a, Min will reply with a strategy that never plays the action α from 2 and therefore making sure that the game never reaches state f. But it is critical for Max to choose the right time to play the letter b. Because if he plays it when the game is in state i then he loses, since the game goes to the sink state s and 4. But if he plays it when the game is in state c then he wins by going to the state f and 3. Of course he cannot tell when he is in state c, because he has no information. Consequently, no matter what strategy Max chooses, Min can refute it, by acting as follows. If the game is in state 2, and in the next turn Max will play an a, Min plays β, if in the next turn Max will play a b, Min plays α, making sure that with probability $3/4$ the game will end up in the sink state s. Note that we are fixing the strategy of Max first in this analysis, *i.e.* we are considering $\underline{val}(i)$, and we see that it is bounded above by $3/4$. This is because when Max plays ab, and Min follows the strategy that we described above, the state f is reached with probability $3/4$, but if Max plays at least two a's before the b, then f is reached only with $1/4$. Hence $\underline{val}(i) = 3/4$.

The refutation of Min depends on the word that was chosen by Max, in order to refute it we need to know whether in the next turn Max will play an a or a b. If we fix a strategy for Min first, then Max can reach the set of final states with a chance that is strictly larger than $3/4$. This is because if the strategy that is chosen by Min, in some turn will play the action β almost surely, then Max takes advantage of this and plays b at this turn. Otherwise if there is some nonzero probability that the strategy of Min will play the action α at every turn, this adds up over time, making the chance of going to the final state f equal to 1, given that Max plays only a. Thus we see that in general

$$\underline{val}(s) = \sup_{w \in \Sigma_1} \inf_{\tau \in \Sigma_2} \mathbb{P}_s^{w,\tau}(\mathbf{F}) \neq \inf_{\tau \in \Sigma_2} \sup_{w \in \Sigma_1} \mathbb{P}_s^{w,\tau}(\mathbf{F}).$$

Nevertheless we will consider only the maxmin for now. △

4.2.3 *Deterministic strategies for Min suffice*

Min is advantaged in two ways. First, we are considering the maxmin value, so we fix first the strategy of Max and then Min replies, knowing this strategy. Second, Max uses a very particular and simple strategy, a pure finite word. With this in mind, it is not surprising that a general behavioral strategy for Min is not necessary. Min can manage with a simpler type of strategy, that is a deterministic strategy that depends on the current state of the game, as well as which turn it is. We denote the set of such strategies by Σ_2^p, it consists of strategies that map

$$\mathbb{N} \to (\mathbf{S_2} \to \mathbf{A}_2),$$

i.e. strategies that at every turn play according to some pure and memoryless strategy.

We prove that Min cannot gain more by choosing some strategy that is in Σ_2 but not in Σ_2^p.

Lemma 4.6.
Let $\mathcal{G}$ be some game with the set of final states $\mathbf{F}$, and $s \in \mathbf{S}$, then

$$\underline{val}(s) = \sup_{w \in \Sigma_1} \inf_{\tau \in \Sigma_2^p} \mathbb{P}_s^{w,\tau}(\mathbf{F}) = \sup_{w \in \Sigma_1} \inf_{\tau \in \Sigma_2} \mathbb{P}_s^{w,\tau}(\mathbf{F}).$$

Proof.
After fixing a finite word $w \in \Sigma_1$ and state s, we can construct a Markov decision process (MDP) with $n \cdot |\mathbf{S_2}|$ states, where n is the length of w as follows. Let

$w = a_1 \cdots a_n$ where $a_i \in \mathbf{A_1}$, the set of states of the MDP is $\{1, \ldots, n\} \times \mathbf{S_2}$. If the initial state s is in $\mathbf{S_1}$, then the initial distribution of the MDP is the same as $p(s, a_1)$ but over the states $\{1\} \times \mathbf{S_2}$. With an action from a state in $\{1\} \times \mathbf{S_2}$ we can only go to a state in $\{2\} \times \mathbf{S_2}$ (by taking into account that Max plays a_2 as a reply), and so on, in general from a state in $\{i\} \times \mathbf{S_2}$ we go to a state in $\{i+1\} \times \mathbf{S_2}$, for the states in $\{n\} \times \mathbf{S_2}$, with every action we loop. In this MDP, Min wants to stay away from the states $\{n\} \times \mathbf{F}$, and he can do this with a pure and memoryless strategy [Puterman, 1995]. A pure and memoryless strategy in this MDP translates to a strategy in Σ_2^p in the half-blind game. ∎

As a consequence of this, in the sequel, we will assume that Min chooses a strategy in Σ_2^p, which will simplify the proofs.

4.2.4 *The belief monoid algorithm*

Given a half-blind game, we will construct an algebraic structure called the belief monoid. The decision procedure consist of this construction and then searching for a particular element in the monoid. If the game belongs to the class of leaktight games then this abstraction using the belief monoid will be faithful. We are going to first define the belief monoid, as well as the belief monoid algorithm, illustrate a few examples, and then later on we will give a precise definition of the class of leaktight half-blind games.

The belief monoid relies on the Markov monoid of [Fijalkow *et al.*, 2012], defined in Definition 3.16. In a sense it is a nesting of the Markov monoid. First, given a half-blind game, we can construct its Markov monoid. In the case of probabilistic automata we were abstracting the finite set of the stochastic matrices M_a by the binary matrices B_a, by mapping nonzero entries to 1. In the case of half-blind games we will abstract the stochastic matrices $M_{a,\tau}$, where τ is a pure and memoryless strategy. These are $|\mathbf{S_1}| \times |\mathbf{S_1}|$ stochastic matrices that are defined as follows, for all $s, s' \in \mathbf{S_1}$

$$M_{a,\tau}(s, s') = \sum_{t \in \mathbf{S_2}} p(s, a)(t) \cdot p(t, \tau(t))(s').$$

In other words $M_{a,\tau}(s, s')$ gives the probability to go from state s to state s' in the game that takes only one turn, and where Max plays the single letter a and Min replies with the pure and memoryless strategy τ. Observe that if Max chooses some word $w = a_1 \cdots a_n$ and Min chooses some strategy $\tau \in \Sigma_2^p$,

$$\mathbb{P}_s^{w,\tau}(s') = \left(M_{a_1,\tau(1)} M_{a_1,\tau(2)} \cdots M_{a_n,\tau(n)}\right)(s, s').$$

For $a \in \mathbf{A}_1$ and τ pure and memoryless (*i.e.* an element of $\mathbf{S_2} \to \mathbf{A}_2$), we define the binary matrices (that abstract the stochastic ones) by

$$B_{a,\tau}(s, s') = 1 \iff M_{a,\tau}(s, s') > 0. \tag{49}$$

Now we can define the Markov monoid for half-blind games as one would assume, considering the previous chapter.

Definition 4.7 (Markov monoid for half-blind games).
The Markov monoid for half-blind games $\mathcal{M}$ is defined as

$$\mathcal{M} = \langle \{ B_{a,\tau} \mid a \in \mathbf{A}_1, \tau \in \mathbf{S_2} \to \mathbf{A}_2 \} \cup \{ I \} \rangle,$$

where I is the unit matrix.

We recall that for a set of binary matrices $\mathbf{b}$, we said $\langle \mathbf{b} \rangle$ is the smallest set that contains $\mathbf{b}$ and is closed under taking products and iterations. Where the last two are defined in Definition 3.8 and Definition 3.11 respectively.

The elements of $\mathcal{M}$ say something about how the game can progress. For example, consider

$$B = B_{a_1,\tau(1)} \cdots B_{a_n,\tau(n)},$$

where $\tau \in \Sigma_2^p$ and $a_1 \cdots a_n \in \Sigma_1$. If for some $s, s' \in \mathbf{S_1}$, $B(s,s') = 1$, then the chance of reaching s' from s when Max plays the word $a_1 \cdots a_n$ and Min the strategy τ, is nonzero. Or, for example, if we have some idempotent $B_{a,\tau} \in \mathcal{M}$, $a \in \mathbf{A}_1$, τ memoryless and pure, and $s, s' \in \mathbf{S_1}$ such that for the iterated element $B^{\#}_{a,\tau}(s,s') = 0$, then this means that

$$\lim_n \mathbb{P}_s^{a^n,\tau}(s') = 0.$$

That is, if Max plays longer and longer chain of a's and Min replies always with the same memoryless and pure strategy τ, then the chance of going from state s to state s' tends to zero.

We need some more structure in the belief monoid, the elements of $\mathcal{M}$ need to be grouped according to what *might* happen when Max chooses some strategy.[1]

The elements of the belief monoid are subsets of $\mathcal{M}$. We group together those elements which correspond to the same strategy by Max. For instance for $a \in \mathbf{A}_1$,

$$\mathbf{a} = \{ B_{a,\tau} \mid \tau \in \mathbf{S_2} \to \mathbf{A}_2 \}. \tag{50}$$

We will denote elements of the belief monoid by boldfaced lowercase letters. The element $\mathbf{a}$ groups together all that might happen in the game when Max

1 even though strategies of Max are finite and pure words, we will sometimes informally use the same term to mean sequences of finite and pure words, since we are dealing with the maxmin reachability problem, which is a question about the asymptotic behavior

plays the letter a. The product of two elements of the belief monoid is just the product of its elements.

Definition 4.8 (Product operator for the belief monoid).
The product of $\mathbf{u}$ and $\mathbf{v}$, denoted by $\mathbf{uv}$ is defined as

$$\mathbf{uv} = \{UV \mid U \in \mathbf{u}, V \in \mathbf{v}\}.$$

Now, for example, to any finite word $w = a_1 a_2 \cdots a_n$, we can associate the element $\mathbf{w} = \mathbf{a}_1 \mathbf{a}_2 \cdots \mathbf{a}_n$, that consists of binary matrices that together encompass all that is possible to occur in the game when Max plays the word w.

As with probabilistic automata, this is not the end of the story, since we need to take into account that in the limit Max can make the probability to go to certain states tend to zero. For this purpose, we define the iteration operator for the belief monoid as follows.

Definition 4.9 (Iteration operator for the belief monoid).
Let $\mathbf{u}$ be idempotent, its iteration, denoted by $\mathbf{u}^{\#}$, is defined as

$$\mathbf{u}^{\#} = \langle \{UE^{\#}V \mid U, E, V \in \mathbf{u}, E^2 = E\} \rangle.$$

In $\mathbf{u}^{\#}$, we are taking elements of $\mathbf{u}$ that are iterated at least once, and then closing that set under product and iteration.

Having defined these two operations for subsets of $\mathcal{M}$, we are now ready to define the belief monoid.

Definition 4.10 (The belief monoid).
Given a half-blind game $\mathcal{G}$, the belief monoid that is associated to it, denoted by $\mathcal{B}$, is the smallest set that contains

$$\{\mathbf{a} \mid a \in \mathbf{A}_1\} \cup \{\{I\}\},$$

and is closed under taking product and iteration, where I is the unit matrix and $\mathbf{a}$ is defined as in (50).

Example 14.
We go back to the game in Figure 13. Min has two pure and memoryless strategies in total, the one that plays α and the one that plays β. We construct its belief monoid. By definition we have

$$B_{a,\alpha} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \textit{and}$$
$$B_{a,\beta} = \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}.$$

As well as,

$$\mathbf{a} = \{B_{a,\alpha}, B_{a,\beta}\}.$$

We have used the order $i < f$ of the states for the matrices above, *e.g.* the $(1,1)$ entry of $B_{a,\alpha}$ corresponds to (i,i), the probability to loop in the state i.

Observe that the set $\mathbf{a}$, is closed under taking products, hence $\mathbf{a}^2 = \mathbf{a}$, *i.e.* $\mathbf{a}$ is idempotent and $\mathbf{a}^\#$ is well-defined. Moreover both of the elements $B_{a,\alpha}$ and $B_{a,\beta}$ are idempotent, and $B_{a,\beta}$ is absorbing, in the sense that

$$B_{a,\alpha}B_{a,\beta} = B_{a,\beta}B_{a,\alpha} = B_{a,\beta}.$$

Since $B_{a,\alpha}^\# = B_{a,\beta}$ we conclude that

$$\mathbf{a}^\# = \{B_{a,\beta}\}.$$

Furthermore, $\mathbf{a}$ and $\mathbf{a}^\#$ are the only elements of $\mathcal{B}$, the belief monoid that is associated to the game in Figure 13. △

In the example above the element $\mathbf{a}^\#$ serves as a *reachability witness*. It witnesses the fact that the state f can be reached with probability arbitrarily close to 1, from the state i.

Definition 4.11 (Maxmin reachability witness).
Let $\mathcal{G}$ be a half-blind game with initial state i and set of final states $\mathbf{F}$, and let $\mathcal{B}$ be the belief monoid that is associated to it. An element $\mathbf{u} \in \mathcal{B}$ is called a *maxmin reachability witness* if for all $B \in \mathbf{u}$, and $s \in \mathbf{S_1}$,

$$B(i,s) = 1 \implies s \in \mathbf{F}.$$

The belief monoid algorithm, constructs the belief monoid and then searches for a maxmin reachability witness. If it finds one, then it returns yes, otherwise it returns no.

Roughly speaking, elements $\mathbf{u} \in \mathcal{B}$ of the belief monoid correspond to strategies of Max, while elements $B \in \mathbf{u}$ inside it, correspond to a response strategy for Min. In this sense the maxmin reachability witness says that there exists some element $\mathbf{u} \in \mathcal{B}$ (some strategy for Max), such that for all $B \in \mathcal{B}$ (for all strategies for Min), the result of the game is that from the initial state we end up in the set of final states.

Algorithm 1: The belief monoid algorithm.

Data: A leaktight half-blind game.
Result: Answer to the Maxmin reachability problem.
$\mathcal{B} \leftarrow \{\mathbf{a} \mid a \in \mathbf{A}_1\}$.
Close $\mathcal{B}$ by product and iteration
Return true iff there is a reachability witness in $\mathcal{B}$

Example 15.
Let us go back to the game in Figure 17 and construct its belief monoid. We use the following order on the states: $i < c < s < f$.

$$\mathbf{a} = \{B_{a,\alpha} = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, B_{a,\beta} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}\}$$

$$\mathbf{b} = \{B_{b,\alpha} = B_{b,\beta} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}\}$$

Moreover $\mathbf{b} = \mathbf{b}^2 = \mathbf{b}^{\#}$. On the other hand, observe that $\mathbf{a}^2 \neq \mathbf{a}$. The elements of $\mathbf{a}^2$ are as follows.

$$B_{a,\alpha} \cdot B_{a,\alpha} = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \qquad B_{a,\alpha} \cdot B_{a,\beta} = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

$$B_{a,\beta} \cdot B_{a,\beta} = B_{a,\beta}, \qquad B_{a,\beta} \cdot B_{a,\alpha} = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

By computing $\mathbf{a}^4$, it follows that $\mathbf{a}^2$ is idempotent, therefore $(\mathbf{a}^2)^\#$ is well-defined. Furthermore, $(\mathbf{a}^2)^\#$ is a strict superset of $\mathbf{a}^2$, containing the following new element:

$$(B_{a,\alpha} \cdot B_{a,\alpha})^\# = (B_{a,\alpha} \cdot B_{a,\beta})^\# = (B_{a,\beta} \cdot B_{a,\alpha})^\# = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Intuitively this element is saying that if Max keeps playing a longer and longer sequence of a^2, and Min replies to each one with (α, α), (α, β) or (β, α), then the final state is reached almost surely. This is because playing α by Min ensures that $1/4$ of the probability that is not in f goes to f, and since f is a sink state, it remains there.

From here, without continuing the computation, it is easy to conclude that $(\mathbf{a}^2)^\#\mathbf{b}$ is not a maxmin reachability witness, since it contains the element

$$B = (B_{a,\beta}^2)^\# \cdot (B_{a,\beta} \cdot B_{a,\alpha}) \cdot B_{b,\alpha} = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

and $B(i, s) = 1$, but $s \notin \mathbf{F}$. Other elements of $\mathcal{B}$ are easily seen to not be maxmin reachability witnesses either. △

4.2.5 *The extended belief monoid*

For the proofs of the correctness of the belief monoid algorithm, as well as the definition of the leaktight half-blind games we will make use of the *extended belief monoid*. This is just the belief monoid that is built on top of the extended Markov monoid that was defined in Definition 3.15.

Definition 4.12 (Extended belief monoid).
The extended belief monoid, denoted by $\widetilde{\mathcal{B}}$, has as elements subsets of $\widetilde{\mathcal{M}}$, otherwise it is defined just as the belief monoid, *i.e.* it is the smallest set that contains

$$\{\mathbf{a} \mid a \in \mathbf{A}_1\} \cup \{\{(I, I)\}\},$$

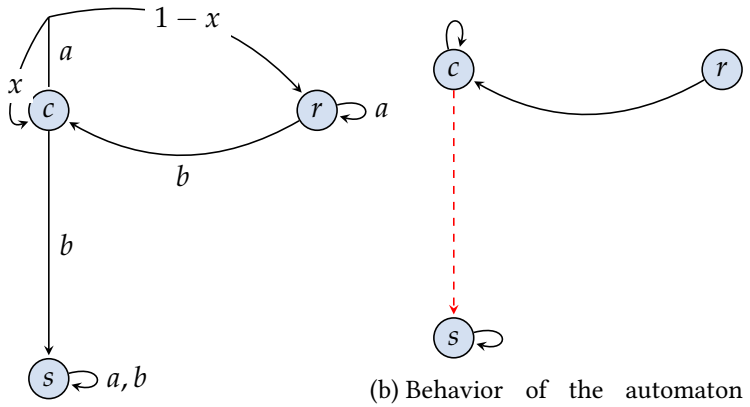
and is closed under product and iteration, where $\mathbf{a}$ are defined as

$$\mathbf{a} = \{(B_{a,\tau}, B_{a,\tau}) \mid \tau \in \mathbf{S_2} \to \mathbf{A}_2\}.$$

4.3 LEAKS

The concept of leaks was introduced in [Fijalkow *et al.*, 2012], as we have discussed in the previous chapter it is interesting for various reasons. In this chapter we will give an informal description of leaks, as well as define them for half-blind games. See [Fijalkow *et al.*, 2012] and [Fijalkow *et al.*, 2015] for examples, and further discussions on leaks.

Let us consider the simplest possible example that exhibits leaks.



(a) A probabilistic automaton that has a leak

(b) Behavior of the automaton when reading the sequence of words $(a^{f(n)}b)_{n\in\mathbb{N}}$

Figure 18: Illustrative example with a leak

In Figure 18a we have a probabilistic automaton with two letters, where x is some rational with $0 < x < 1$. We regard the behavior of the automaton when it reads the sequence of words $((a^{f(n)}b)^{g(n)})_{n\in\mathbb{N}}$, where f and g are some increasing functions. If we start from the state c and play the sequence $a^{f(n)}b$, the chance of coming back to c is $1 - x^{f(n)}$. This is because the chance of going to the sink state s, is $x^{f(n)}$, and the probability to go to the state r is 0. So with a very small probability we are going to the sink state, otherwise we come back to c. Up until now, the behavior of the automaton is fairly straight forward. The complications arise when we repeat the word $a^{f(n)}b$. There is some small chance that we go to the sink state s, but we repeat this experiment $g(n)$ times, thereby increasing the probability of being stuck in the sink state. Will there always be some nonzero chance of going to the sink state in the limit, is hard to ascertain, and in fact it is algorithmically impossible in the general case, since this is a leak. In our case here, the answer depends on the quantity x as well as how fast the functions f and g increase; so already it depends on the quantitative properties of the automaton, and not just the qualitative ones.

Under the sequence of words $(a^{f(n)}b)_{n\in\mathbb{N}}$ the state c is recurrent and it forms its own recurrence class, the same holds for s. We have illustrated this in Figure 18b. By this we mean that in the limit of that sequence of words, there is no nonzero probability of going from the state c to some other state and not coming back (note that the dotted transition from c to s tends to zero in the limit).

While the dotted transition is zero in the limit, it is nonzero for all $n \in \mathbb{N}$. What we are dealing with here, is two recurrence classes that still communicate with each other with some transition that has little chance of occurring. This is what we refer to as a leak, the transition between the two recurrence classes that has little chance of occurring, the recurrence class $\{r\}$ *leaks* some probability to the recurrence class $\{s\}$.

We summarize this discussion in the figure below.

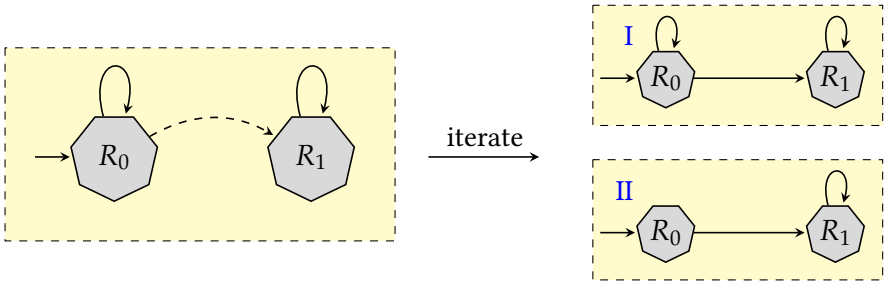


Figure 19: Summary of leaks

We have two recurrence classes, R_0 and R_1, and R_0 leaks some probability to R_1. By repeating, or iterating this many times there are two possible things that might happen, either some probability remains in R_0 (this is (I) in Figure 19) or the leak vanishes too slowly and therefore depletes all the probability from R_0 (this is (II) in Figure 19).

Which one of these two possible outcomes happens is algorithmically too complicated to conclude, and in general impossible. It becomes even more complicated when there are two or more leaks at the same time, and we have to compare their speeds. See [Fijalkow *et al.*, 2012] for such examples, and [Fijalkow, 2015] for further discussion on the speeds of convergence of leaks.

Hopefully the discussion is sufficient to motivate the syntactic characterization, which serves as a definition, of leaks given in Definition 3.16. We recall this definition here, for the extended Markov monoid that is associated to a half-blind game.

Definition 4.13 (Leaktight half-blind games).
Let $\mathcal{G}$ be a half-blind game and $\widetilde{\mathcal{M}}$ the extended Markov monoid that is associated to it. An idempotent element $(B, \widetilde{B}) \in \widetilde{\mathcal{M}}$ is called a *leak* if there exists $r, r' \in \mathbf{S_1}$ such that

- r and r' are B-recurrent,
- $B(r, r') = 0$,
- $\widetilde{B}(r, r') = 1$.

If $\widetilde{\mathcal{M}}$ does not contain a leak then we say that the game $\mathcal{G}$ is leaktight.

What follows from here is that in order to ascertain whether a given game is leaktight, it suffices to check if any element of its extended Markov monoid fulfills the properties given in Definition 4.13, so it is decidable.

Theorem 4.14.
Given a half-blind game, one can decide whether it is leaktight.

This is a very convenient property of this subclass of automata and half-blind games: having a class of games with decidable maxmin reachability problem whose membership we cannot decide would not be very useful.

Example 16.
In the extended Markov monoid of the automaton in Figure 18a, we have the following couple(we have used the order $c < r < s$ on the states),

$$B_a^{\#} B_b = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \widetilde{B_a^{\#} B_b} = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

In the right component we have kept the transition from c to s. Observe that this element fulfills all the properties in Definition 4.13 for the states c and s, and therefore is a leak. △

We have illustrated above the leaks on probabilistic automata instead of half-blind games for the sake of simplicity. The situation for games is very similar, in fact, it is easy to see that, given a game $\mathcal{G}$ we can construct an automaton $\mathbb{A}$

such that $\mathcal{G}$ is leaktight if and only if $\mathbb{A}$ is. The automaton has an alphabet of size $|\mathbf{A}_1| \times |\mathbf{S_2} \to \mathbf{A}_2|$.

It is true that in games the asymptotic behavior can be much more complex, since Min can choose some complex strategy, but as we shall see, when the game is leaktight, none of such complex strategies help Min to impede Max from reaching the final states, and hence do not make the question of maxmin reachability harder.

4.4 k-DECOMPOSITION TREES

The notion of k-decomposition trees is central in the proofs of the correctness of the belief monoid algorithm. We will define k-decomposition trees in this section, and prove a key property that they have: a bound on their height.

The setting is as follows. We have a finite alphabet A, a finite monoid $(M, \cdot)$ and a morphism ϕ from A^* (the set of finite words with alphabet A) to M. This means that for all $u, v \in A^*$, we have $\phi(uv) = \phi(u) \cdot \phi(v)$.

Since the set A^* is infinite but M is finite, Ramsey's theorem tells us that for all $n \in \mathbb{N}$, any sufficiently long word $w \in A^*$ can be decomposed into factors $w = uw_1w_2 \cdots w_nv$ such that $\phi(w_1) = \phi(w_2) = \cdots \phi(w_n)$.

Simon's factorization forest theorem [Simon, 1990], is a strong extension of Ramsey factorization. There we have decomposed the word into factors, Simon's theorem completely decomposes the word into a tree whose height is independent of the length of the word. More precisely it decomposes a word $w \in A^*$ into a Ramsey decomposition tree of w. This is a rooted tree whose nodes are labeled by pairs in $A^* \times M$ that follows the following rules:

- the leaves are labeled by $(a, \phi(a))$, where $a \in A$,
- the product nodes have exactly two children, if the children are labeled by (u_1, m_1) and (u_2, m_2) then the product node, their parent, is labeled by $(u_1u_2, m_1 \cdot m_2)$,
- the idempotent nodes have more than two children whose label's right component coincides and is idempotent, if the children are labeled by $(u_1, e), (u_2, e), \ldots, (u_n, e)$, where e is idempotent, then the idempotent node, their parent, is labeled by $(u_1 \cdots u_n, e)$,
- the root is labeled by $(w, \phi(w))$.

A tree whose every node is either a product, an idempotent node or a leaf and that fulfills the rules above is a Ramsey decomposition tree.

Theorem 4.15 ([Simon, 1990]).
For all $w \in A^*$ there exists a Ramsey decomposition tree with height at most $9|M|$.

The k-decomposition trees, $k \in \mathbb{N}$, are analogues of the Ramsey decomposition tree for the monoids that are equipped with the unary operator (#) as well as the product. Instances of such monoids $(M, \cdot, \#)$ where # maps $e(M)$ to $e(M)$, $e(M)$ being the set of idempotents of M, are the Markov and belief monoids, among others.

In [Simon, 1994] and [Fijalkow *et al.*, 2012] 2-decomposition trees were used in conjunction with a similar theorem to Theorem 4.15. The notion of a k-decomposition tree was introduced in [Colcombet, 2013].

We define k-decomposition trees. Let A be a finite alphabet, and $(M, \cdot)$ a monoid equipped with a unary operator # that maps $e(M)$ to $e(M)$, $e(M)$ being the set of idempotents of M.

Definition 4.16 (k-decomposition tree).
Given $w \in A^*$, and $k \in \mathbb{N}$, $k > 1$, a k-decomposition tree of w is a rooted tree whose nodes are labeled by elements of $A^* \times M$ that follows the following rules:

- the leaves are labeled by $(a, \phi(a))$, $a \in A$,
- the product nodes have exactly two children, if the children are labeled by (u_1, m_1) and (u_2, m_2) then the product node, their parent, is labeled by $(u_1 u_2, m_1 \cdot m_2)$,
- the idempotent nodes have at most k children whose label's right component coincides and is idempotent, if the children are labeled by $(u_1, e), (u_2, e), \ldots, (u_n, e)$, with e idempotent and $n \leq k$, then the idempotent node, their parent, is labeled by $(u_1 \cdots u_n, e)$,
- the iteration nodes have at least $k+1$ children whose label's right component coincides and is idempotent, if the children are labeled by $(u_1, e), (u_2, e), \ldots, (u_n, e)$, where e is idempotent and $n > k$, then the iteration node, their parent, is labeled by $(u_1 \cdots u_n, e^{\#})$.
- the root node is labeled by (w, m) for some $m \in M$.

In [Fijalkow *et al.*, 2012], 2-decomposition trees are used, *i.e.* there are no idempotent nodes, only product and iteration nodes. The reason why we use the more general notion of the k-decomposition tree for half-blind games is roughly as follows.

In the case of probabilistic automata, the 2-decomposition trees are used to prove a lower bound on the probability to go from a state to another state. In the proof we inductively walk up the tree, and use the fact that there exists a bound on the height of the tree that does not depend on the length of the word.

In the case of half-blind games, we will use decomposition trees to prove some upper bounds as well, and for this we need to make sure that the chance to go to the transient states decreases in the game. This comes as a result of iterating some word many times, but in a 2-decomposition tree, an iteration node can have as little as three children, there is no lower bound on the number of children and consequently we cannot gain an upper bound that decreases on the chance to go to a transient state. For this reason we need more control on the number of children of iteration nodes and one way to achieve it is through k-decomposition trees.

Example 17.
Let $A = \{a, b\}$, $(M, \cdot)$ some monoid with (#) a unary operator that maps $e(M)$ to $e(M)$, and ϕ morphism from A^* to M such that $\phi(a)$ and $\phi(b)$ are idempotent. A 4-decomposition tree of the word *aaaaabbbb* is depicted in Figure 20.

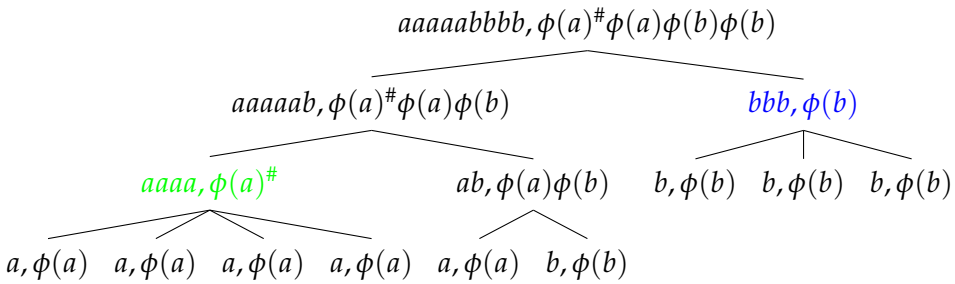


Figure 20: A 4-decomposition tree of the word *aaaaabbbb*

Observe that the node labeled by $bbb, \phi(b)$ is an idempotent node, whereas the one that is labeled by $aaaa, \phi(a)^{\#}$ is an iteration node. △

In this section we are going to prove an analogue of Theorem 4.15 for the k-decomposition trees and monoids equipped with a unary operator (#) that has the following properties:

$$\forall e \in e(M), e^{\#} \in e(M) \tag{51}$$
$$\forall e \in e(M), (e^{\#})^{\#} = e^{\#} \tag{52}$$
$$\forall e \in e(M), ee^{\#} = e^{\#}e = e^{\#}. \tag{53}$$

The analogue result for 2-decomposition trees, with a proof in the same spirit as the one given here, can be found in [Simon, 1994; Fijalkow *et al.*, 2012]. For k-decomposition trees, a proof can be found in [Colcombet, 2009], but it asks for an additional property for the monoid. It asks that $(ab)^\# a = a(ba)^\#$ where $ab, ba \in e(M)$. We require only (51), (52) and (53).

Lemma 4.17.
Let A be a finite set, $(M, \cdot)$ a monoid that is equipped with a unary operator $\# : e(M) \to e(M)$, such that (51), (52) and (53) hold, and ϕ a morphism from A^* to M. For all $w \in A^*$, $k > 1$, there exists a k-decomposition tree of w whose height is at most $9|M|^2$.

Before we prove Lemma 4.17, we have to introduce some essential finite semigroup results. In particular Green's relations, which are the main tool in the study of finite semigroups. These results can be found in any textbook on finite semigroups. See for example [Pin and Miller, 1986] and [Clifford and Preston, 1961].

Let $u \in M$, we use the following notation

$$\begin{aligned} uM &= \{uu' \mid u' \in M\} \\ Mu &= \{u'u \mid u' \in M\} \\ MuM &= \{vuv' \mid v, v' \in M\}. \end{aligned}$$

We will define the relations $\mathcal{L}, \mathcal{R}, \mathcal{J}, \mathcal{H}$ and $\mathcal{D}$, that are equivalence relations on M. These relations together are called Green's relations.

Definition 4.18 (Green's relations).
Let $u, v \in M$,

- $u \mathcal{L} v \iff Mu = Mv$,
- $u \mathcal{R} v \iff uM = vM$,
- $u \mathcal{J} v \iff MuM = MvM$,
- $u \mathcal{H} v \iff u \mathcal{L} v$ and $u \mathcal{R} v$,
- $u \mathcal{D} v \iff \exists w \in M, u \mathcal{L} w$ and $w \mathcal{R} v$

For $\mathcal{L}, \mathcal{R}, \mathcal{J}$ we can define the partial orders $\leq_{\mathcal{L}}, \leq_{\mathcal{R}}, \leq_{\mathcal{J}}$, for all $u, v \in M$, $u \leq_{\mathcal{L}} v$ if and only if $Mu \subseteq Mv$. Similarly for $\leq_{\mathcal{R}}$ and $\leq_{\mathcal{J}}$. With $<_{\mathcal{L}}, <_{\mathcal{R}}, <_{\mathcal{J}}$ we denote their strict counterparts.

The relations are equivalence relations, so we will use, for example, $\mathcal{J}$-class to mean an equivalence class with respect to the $\mathcal{J}$ relation.

We recall some basic results, whose proofs can be found in the books cited above.

Lemma 4.19.
No $\mathcal{H}$-class contains more than one idempotent element.

Lemma 4.20.
Let $u, v \in M$,

- If $u \leq_{\mathcal{L}} v$ and $u \mathcal{J} v$, then $u \mathcal{L} v$.
- If $u \leq_{\mathcal{R}} v$ and $u \mathcal{J} v$, then $u \mathcal{R} v$.

Now we will give a lemma that is key in the proof of Lemma 4.17, it can be found in [Simon, 1994]. We assume that the monoid M is such that it fulfills the properties (51), (52) and (53).

Lemma 4.21 ([Simon, 1994]).
Let $e \in M$ be an idempotent element such that $e^{\#} \neq e$. Then $e^{\#} <_{\mathcal{J}} e$.

Proof.
First observe that from (53), $ee^{\#}e = e^{\#}$. From this, together with (51), it follows that $e^{\#} \leq_{\mathcal{J}} e$. The proof is by contradiction: assume that $e \mathcal{J} e^{\#}$. Note that $e \leq_{\mathcal{L}} e^{\#} = ee^{\#}$, so from Lemma 4.20, $e \mathcal{L} e^{\#}$. The argument for $e \mathcal{R} e^{\#}$ is dual, and hence $e \mathcal{H} e^{\#}$. Since both e and $e^{\#}$ are idempotent and in the same $\mathcal{H}$-class, from Lemma 4.19 it follows that $e = e^{\#}$, which is a contradiction. ∎

We are now equipped to proceed with the proof of Lemma 4.17.

Proof Idea (of Lemma 4.17).
We start with the Ramsey decomposition tree of some word, and replace nodes that could be iteration nodes (*i.e.* that have at least k children) by some new letter of the alphabet. Each time we do this, because of Lemma 4.21, we descend the $\mathcal{J}$-class, since we can do this at most $|M|$ times, it provides us with a bound. △

Proof (of Lemma 4.17).
Let $w \in A^*$ and $k > 1$. We are going to prove that there exists some k-decomposition tree of w whose height is at most $9J|M|$, where J is the number of $\mathcal{J}$-classes on M. This suffices to conclude the lemma since $J \leq |M|$.

Set $\phi_0 = \phi$ and $A_0 = A$. From Theorem 4.15 there exists a Ramsey decomposition tree T_0 of w whose height is at most $9|M|$. Call any idempotent node of T_0 with children $(u_1, e), (u_2, e), \ldots, (u_n, e)$, *a primitive node*, if $e^\# \neq e$ and $n \geq k$. If T_0 has no primitive node then it is itself a k-decomposition tree and we are done. Otherwise for *all* primitive nodes that are at maximal depth (they have no descendants that are primitive) and that are labeled (u, e) with children $(u_1, e), \ldots, (u_n, e)$, where $n \geq k$, add a new letter a_u to the alphabet A_0, and call this new alphabet A_1. Define the morphism ϕ_1, whose domain is A_1^*, as $\phi_1(a_u) = e^\#$ and $\phi_1(u') = \phi_0(u')$, for all other $u' \in A_0^*$. Transform the word w by replacing the factor u with the new letter a_u. Do the above in parallel for all the primitive nodes that have maximal depth, *i.e.* we possibly add more than one new letter in A_1.

Now we apply again Theorem 4.15 to the alphabet A_1 and the morphism ϕ_1. This gives us a new Ramsey decomposition tree T_1, where the subtree of T_0 whose root has the label (u, e) is replaced by the leaf $(a_u, e^\#)$. If T_1 has no primitive nodem then we unwrap the leaves labeled $(a_u, e^\#)$, by plugging in the tree that was in T_0, except that for its root we use the label $(u, e^\#)$, and we are done, T_1 is a k-decomposition tree. If, on the other hand T_1 still has some primitive nodes, then we repeat the procedure above. Since this procedure replaces more and more factors of w by letters, it must stop at some T_j.

We claim that $j \leq J$. To see this, observe first that for all $u, v, e \in M$ with e idempotent we have $ue^\#v \leq_{\mathcal{J}} e^\#$, $ue^\# \leq_{\mathcal{J}} e^\#$, and $e^\#v \leq_{\mathcal{J}} e^\#$. So when we multiply $e^\#$ with some other element of the monoid we do not ascend in the $\mathcal{J}$-classes. But on the other hand according to Lemma 4.21 whenever we pass from e to $e^\#$, we descend on the $\mathcal{J}$-classes, and since we are doing this with every pass of the procedure above, it follows that $k \leq J$ where J is the number of $\mathcal{J}$-classes. ■

We are going to use Lemma 4.17 in the sequel by instantiating it for the monoids $\widetilde{\mathcal{M}}$ and $\widetilde{\mathcal{B}}$. In order to do so we must make sure that these two monoids fulfill (51), (52), and (53).

Lemma 4.22 ([Fijalkow *et al.*, 2015]).
For the extended Markov monoid (51), (52), and (53) hold.

Lemma 4.23.
For the extended belief monoid (51), (52), and (53) hold.

Proof.
We prove it for the belief monoid $\mathcal{B}$, for the extended version, the proof is the same.

First (51) follows from Definition 4.9: $\mathbf{u}^\#$ is a subset of $\mathcal{M}$ that is closed under taking product, hence $\mathbf{u}^\#\mathbf{u}^\# \subseteq \mathbf{u}^\#$. To see the inverse inclusion observe that $UE^\#V = (UE^\#E)(EE^\#V)$ from Lemma 4.22. Furthermore membership in $\mathbf{u}^\#\mathbf{u}^\#$ remains true under taking products and iterating.

The property (52) is trivial since from Lemma 4.22, for all idempotent $E \in \mathcal{M}$, $(E^\#)^\# = E^\#$, so $(\mathbf{u}^\#)^\#$ has the elements $UE^\#(E^\#)^\#E^\#V = UE^\#V$.

Last the property (53) can be easily proved after making the observation that $UE^\#V = U(EE^\#V)$, which comes from Lemma 4.22. ∎

We now have at our disposal a powerful tool which we will use to demonstrate the correctness of the belief monoid algorithm. In the case of the soundness of the algorithm we will apply Lemma 4.17 to the alphabet $\mathbf{A}_1 \times (\mathbf{S_2} \to \mathbf{A}_2)$ and the corresponding morphism to the Markov monoid that is associated to the game, whereas in the case of the completeness of the algorithm we will apply the lemma to the alphabet $\mathbf{A}_1$ with the corresponding morphism to the Belief monoid that is associated to the given game.

4.5 SOUNDNESS

The proof of correctness of the Belief monoid algorithm is in two parts. The soundness that justifies the *yes* replies of the algorithm and the correctness that justifies the *no* replies of the algorithm.

This section is devoted to demonstrating the following theorem.

Theorem 4.24 (Soundness).
Given a leaktight half-blind game, if its belief monoid has a maxmin reachability witness, then the set of final states is maxmin reachable, that is, $\underline{val}(s) = 1$, where s is the initial state.

This theorem is a corollary of the following lemma.

Lemma 4.25.
Assume that we are given a leaktight half-blind game with belief monoid $\mathcal{B}$. With every $\mathbf{u} \in \mathcal{B}$ we can associate some $(u_n)_{n\in\mathbb{N}}$, $u_n \in \mathbf{A}_1^*$, such that for all sequence $(\tau_n)_{n\in\mathbb{N}}$, $\tau_n \in \Sigma_2^p$, there exists $U \in \mathbf{u}$, and $((u'_n, \tau'_n))_{n\in\mathbb{N}}$ a subsequence of $((u_n, \tau_n))_{n\in\mathbb{N}}$ for which

$$U(s,t) = 0 \implies \lim_n \mathbb{P}_s^{u'_n,\tau'_n}(t) = 0,\ s,t \in \mathbf{S_1}.$$

This lemma materializes the intuition that we gave before, that the strategy choice of Max is selecting some $\mathbf{u} \in \mathcal{B}$, whereas that of Min is selecting some $U \in \mathbf{u}$.

Lemma 4.25 implies Theorem 4.24 because to the maxmin reachability witness $\mathbf{u}$, we associate some sequence of words $(u_n)_{n\in\mathbb{N}}$ such that against any $(\tau_n)_{n\in\mathbb{N}}$ there exists some subsequence $((u'_n, \tau'_n))_{n\in\mathbb{N}}$ of $((u_n, \tau_n))_{n\in\mathbb{N}}$, such that

$$\lim_n \mathbb{P}_s^{u'_n,\tau'_n}(t) = 0,$$

where s is the initial state and $t \notin \mathbf{F}$. This is because from Definition 4.11, for all $U \in \mathbf{u}$, and $t \notin \mathbf{F}$, $U(s,t) = 0$.

Before we give the proof idea of Lemma 4.25, we are going to give a couple of definitions.

The pair of strategies $u \in \mathbf{A}_1^*$ and $\tau \in \Sigma_2^p$ induce a finite word over the alphabet

$$\mathfrak{A} = \{(a, \alpha) \mid a \in \mathbf{A}_1, \alpha \in \mathbf{S_2} \to \mathbf{A}_2\}.$$

In this case, if $u = a_1 \cdots a_n$, it is the word $(a_1, \tau(1)), (a_2, \tau(2)), \ldots, (a_n, \tau(n)) \in \mathfrak{A}^*$. We are going to handle words over $\mathfrak{A}$, and the morphism ϕ to the finite monoid $\mathcal{M}$ (or $\widetilde{\mathcal{M}}$) that is defined by $\phi((a, \alpha)) = B_{a,\alpha}$, where the latter's definition is given in (49) on Page 112.

In a k-decomposition tree, the right components of labels (*i.e.* the elements of the monoid), are called the *types*. The type of the tree is the type of its root. For example the type of the 4-decomposition tree in Figure 20 is $\phi(a)^\#\phi(a)\phi(b)\phi(b) = \phi(a)^\#\phi(a)\phi(b)$.

Definition 4.26.
Let $p \in \mathfrak{A}^*$, $h, k \in \mathbb{N}$. We denote by $T_k^h(p)$, the set of all k-decomposition trees of p whose height is at most h.

Whereas by $\mathfrak{T}_k^h(p)$ we denote the set of all types of trees that appear in $T_k^h(p)$.

Lemma 4.17 says that there exists some k-decomposition tree of certain height, but there might be more (even of different types), or there might be none if the height is too small.

Definition 4.27 (Realization).
Let $(p_n)_{n\in\mathbb{N}}$, $p_n \in \mathfrak{A}^*$, $h \in \mathbb{N}$, and $\mathbf{X} \subseteq \widetilde{\mathcal{M}}$. We say that $(p_n)_{n\in\mathbb{N}}$ *realizes* $\mathbf{X}$ with height h if there exists some subsequence $(p'_n)_{n\in\mathbb{N}}$ of $(p_n)_{n\in\mathbb{N}}$ and $k \in \mathbb{N}$ such that

$$\mathfrak{T}_k^h(p'_n) = \mathbf{X}, n \in \mathbb{N},$$

and for infinitely many $i > k$, $\mathbf{X}$ appears infinitely often in the sequence $(\mathfrak{T}_i^h(p'_n))_{n\in\mathbb{N}}$.

Consider the following table for some $h \in \mathbb{N}$ and $(p_n)_{n\in\mathbb{N}}$:

$$\begin{array}{cccccc} \mathfrak{T}_2^h(p_1) & \mathfrak{T}_2^h(p_2) & \cdots & \mathfrak{T}_2^h(p_n) & \cdots \\ \mathfrak{T}_3^h(p_1) & \mathfrak{T}_3^h(p_2) & \cdots & \mathfrak{T}_3^h(p_n) & \cdots \\ \vdots & \vdots & \ddots & \vdots & \ddots \\ \mathfrak{T}_i^h(p_1) & \mathfrak{T}_i^h(p_2) & \cdots & \mathfrak{T}_i^h(p_n) & \cdots \\ \vdots & \vdots & \ddots & \vdots & \ddots \end{array} \tag{54}$$

The sequence $(p_n)_{n\in\mathbb{N}}$ realizes $\mathbf{X}$ if there exists a row k in the table above with infinitely many elements equal to $\mathbf{X}$, and furthermore, after deleting all the columns with elements that are not equal to $\mathbf{X}$, in all rows below k, $\mathbf{X}$ appears infinitely often. This means that for all $U \in \mathbf{X}$, we can construct a k-decomposition tree of p_n of height at most h, for larger and larger k and n, that has type U. This is useful because we are going to prove an upper bound, as a function of k, on the probability to go from s to t with p_n, whenever $U(s,t) = 0$. Moreover this bound will tend to zero as k tends to infinity. In this way we will demonstrate that the game under $(p_n)_{n\in\mathbb{N}}$ behaves similarly to U. In other words $(p_n)_{n\in\mathbb{N}}$ realizes U, and in general any element of $\mathbf{X}$.

Proof Idea (of Lemma 4.25).
We will prove Lemma 4.25 in two steps. Both proofs are by induction on the structure of the monoid $\mathcal{B}$, that is, the base case is for $\mathbf{a} \in \mathcal{B}$, where $a \in \mathbf{A}_1$, and in the induction step we demonstrate that the property remains true under product and iteration.

First we will show that for all $\mathbf{u} \in \mathcal{B}$ there is some $(u_n)_{n\in\mathbb{N}}$, $u_n \in \mathbf{A}_1^*$, such that against all replies $(\tau_n)_{n\in\mathbb{N}}$, $\tau_n \in \Sigma_2^p$, there exists $\mathbf{X} \subseteq \mathcal{M}$, with $\mathbf{X} \cap \mathbf{u} \neq \emptyset$, and $((u_n, \tau_n))_{n\in\mathbb{N}}$ realizes $\mathbf{X}$.

In particular, this means that we can construct k-decomposition trees of (u_n, τ_n), of height at most h (for some $h \in \mathbb{N}$), for larger and larger n and k whose type is some $U \in \mathbf{u}$.

In the second step we will demonstrate the existence of an upper bound $f(h,k)$ such that if we are given some k-decomposition tree of (u,τ) of height h, whose type is U, then

$$U(s,t) = 0 \implies \mathbb{P}_s^{u,\tau}(t) \leq f(h,k).$$

Moreover $f(h,k)$ tends to zero as k tends to infinity. This we will prove by walking up the tree inductively, that is, we show it for trees that are leaves, and then prove that it remains true for the parent, given that it is true for the children.

The leaktight hypothesis is used when proving the existence of this upper bound, in the case of iteration nodes (when we assume that it is true for the children, and we want to prove that it is true for the parent as well, when the parent is an iteration node). △

We now prove the first step, by showing initially that for a sequence $(p_n)_{n\in\mathbb{N}}$, there exists some $\mathbf{X} \subseteq \mathcal{M}$ that realizes it. This is the following lemma. Its proof is a simple argument whose essential part is that $\mathcal{M}$ is finite.

Lemma 4.28.
Let $(p_n)_{n\in\mathbb{N}}$, $p_n \in \mathfrak{A}^*$. There exists $h \in \mathbb{N}$, and $\mathbf{X} \subseteq \mathcal{M}$, such that $(p_n)_{n\in\mathbb{N}}$ realizes $\mathbf{X}$ with height h.

Proof.
Set $h = 9|\mathcal{M}|^2$. From Lemma 4.17, for all $k > 2$ and $n \in \mathbb{N}$,

$$\mathfrak{T}_k^h(p_n) \neq \emptyset.$$

Since $\mathcal{M}$ is finite, there exists some $\mathbf{X}_1 \subseteq \mathcal{M}$ and some subsequence $(p'_n)_{n\in\mathbb{N}}$ of $(p_n)_{n\in\mathbb{N}}$, such that $\mathfrak{T}_2^h(p'_n) = \mathbf{X}_1$ for all $n \in \mathbb{N}$. If for infinitely many $i > 2$, $\mathbf{X}_1$ appears infinitely often in the sequence $(\mathfrak{T}_i^h(p'_n))_{n\in\mathbb{N}}$, we are done. Otherwise there exists some $k_1 \in \mathbb{N}$ such that for all $i > k_1$, $\mathbf{X}_1$ appears only finitely often in the sequence $(\mathfrak{T}_i^h(p'_n))_{n\in\mathbb{N}}$. Repeating this process, there exists some $\mathbf{X}_2 \subseteq \mathcal{M}$ and a subsequence $(p''_n)_{n\in\mathbb{N}}$ of $(p'_n)_{n\in\mathbb{N}}$ such that $\mathfrak{T}_{k_1}^h(p''_n) = \mathbf{X}_2$ for all $n \in \mathbb{N}$, and so on. The monoid $\mathcal{M}$ being finite implies that this process eventually stops, and there exists some $\mathbf{X} \subseteq \mathcal{M}$ such that $(p_n)_{n\in\mathbb{N}}$ realizes $\mathbf{X}$ with height h. ∎

We now know that all sequences of finite plays (finite words over the alphabet $\mathfrak{A}$) are realizing some $\mathbf{X} \in \mathcal{M}$ for the height $9|\mathcal{M}|^2$. Nevertheless we are going to need something similar having to do with the belief monoid instead of the Markov one. This is the next lemma, which finishes the first step. Its proof is by induction on the structure of $\mathcal{B}$. The other two proofs of this section have the same form.

Lemma 4.29.
For all $\mathbf{u} \in \mathcal{B}$ there exists $(u_n)_{n\in\mathbb{N}}$, $u_n \in \mathbf{A}_1^*$, $h \in \mathbb{N}$, and a function $N : \mathbb{N} \to \mathbb{N}$, such that for all replies $(\tau_n)_{n\in\mathbb{N}}$, $\tau_n \in \Sigma_2^p$, $k > 2$, and $n > N(k)$

$$\mathfrak{T}_k^h((u_n, \tau_n)) \cap \mathbf{u} \neq \emptyset.$$

Above by (u_n, τ_n), we mean the word $(a_1, \tau_n(1)) \cdots (a_l, \tau_n(l))$ over the alphabet $\mathfrak{A}$, where $u_n = a_1 \cdots a_l$.

Proof.
The proof is by induction on the structure of the elements of $\mathcal{B}$. This means that we will prove the lemma first for $\mathbf{a} \in \mathcal{B}$, where $a \in \mathbf{A}_1$ (the base case), then we will assume that the lemma is true for some $\mathbf{u}, \mathbf{v} \in \mathcal{B}$, and prove that it remains true for $\mathbf{uv}$ (product), in the end we will assume that the lemma is true for some idempotent $\mathbf{u} \in \mathcal{B}$ and prove that it remains true for $\mathbf{u}^{\#}$.

Base case. We prove the lemma for all $\mathbf{a} \in \mathcal{B}$, where $a \in \mathbf{A}_1$. Set the sequence of words to be the constant sequence $(a)_{n\in\mathbb{N}}$, $h = 1$ and N the constant function that maps to 0. For all $\tau \in \Sigma_2^p$, and $k > 1$ the unique k-decomposition tree of (a, τ), is the single leaf node whose type is in $\mathbf{a}$ by definition of the morphism ϕ and (50).

Product. Assume that the lemma is true for two elements $\mathbf{u}, \mathbf{v} \in \mathcal{B}$, $(u_n)_{n\in\mathbb{N}}$, h_u, N_u and $(v_n)_{n\in\mathbb{N}}$, h_v, N_v respectively. We will prove that the lemma is true for $\mathbf{uv}$, with the sequence $(u_n v_n)_{n\in\mathbb{N}}$, $h = \max\{h_u, h_v\} + 1$, and $N(k) = \max\{N_u(k), N_v(k)\}$, $k \in \mathbb{N}$.

Let $(\tau_n)_{n\in\mathbb{N}}$ be a sequence of strategies, $k > 2$, and $n > N(k)$. Define

$$\tau'_n(i) = \tau_n(i + |u_n|), n, i \in \mathbb{N},$$

the strategies τ_n that are shifted by the length of the word u_n.

From the induction hypothesis there exists a k-decomposition tree of height at most h_u for (u_n, τ_n) whose root node has some type $U \in \mathbf{u}$, and a k-decomposition tree of height at most h_v for (u_n, τ'_n) whose root node has some type $V \in \mathbf{v}$. Therefore we can construct a k-decomposition tree of height at most $\max\{h_u, h_v\} + 1$ of $(u_n v_n, \tau)$ whose root node has label UV, by making a product node, between whose children are the two corresponding trees. By definition of the product $UV \in \mathbf{uv}$, which concludes this case.

Iteration. Assume that the lemma is true for some idempotent element $\mathbf{u} \in \mathcal{B}$. Then there exists $(u_n)_{n\in\mathbb{N}}$, h_u and N_u for which the lemma holds. We are going to prove that the lemma holds for $\mathbf{u}^{\#} \in \mathcal{B}$, for $(u_n^n)_{n\in\mathbb{N}}$, $h = h_u + 9|\mathcal{M}|^2$ and the function N defined as

$$N(k) = N_u(k) + k^{9|\mathcal{M}|^2}, k \in \mathbb{N}.$$

Let $(\tau_n)_{n\in\mathbb{N}}$, $\tau_n \in \Sigma_2^p$, $k > 2$, and $n > N(k)$ and define

$$\tau_n^i(j) = \tau_n(j + |u_n^i|), i, j, n \in \mathbb{N},$$

the strategies τ_n that are shifted by the length of the word u_n^i. Since $n > N(k) > N_u(k)$, from the induction hypothesis, for all strategies $\tau \in \Sigma_2^p$, $\mathfrak{T}_k^{h_u}((u_n, \tau)) \cap \mathbf{u} \neq \emptyset$, therefore for all $0 \leq i < n$ pick some

$$U_i \in \mathfrak{T}_k^{h_u}((u_n, \tau_n^i)) \cap \mathbf{u}$$

and denote by T_i the corresponding k-decomposition tree. Similarly to the proof of Lemma 4.17, we will modify the alphabet $\mathfrak{A}$, by adding (u_n, τ_n^i), $0 \leq i < n$, as letters, as well as modifying the morphism ϕ so that it maps (u_n, τ_n^i) to U_i. Applying Lemma 4.17 to this new alphabet, morphism, and the word (u_n^n, τ_n) we have that there exists a k-decomposition tree of height at most $9|\mathcal{M}|^2$, where the leaves are labeled by $((u_n, \tau_n^i), U_i)$, $0 \leq i < n$. Unwrapping the trees T_i for the leaves, we construct a k-decomposition tree T of height at most $h = h_u + 9|\mathcal{M}|^2$. Since the number of such leaves is $n > N(k) \geq k^{9|\mathcal{M}|^2}$, T needs to have at least one iteration node in order to be of height smaller than h, therefore the type of T has at least one iteration. It now follows from the definition of $\mathbf{u}^{\#}$, that the type of T must belong to $\mathbf{u}^{\#}$. This concludes the iteration case.

Observe that we have not fixed a height h for which the lemma is true for all $\mathbf{u} \in \mathcal{B}$, but we start with height 0 and increase it by 1 for the product nodes and by $9|\mathcal{M}|^2$ for iteration nodes, but since $\mathcal{B}$ is finite, it follows that we can choose the height $h = 9|\mathcal{M}|^2|\mathcal{B}|$. ∎

The two lemmas above finish up the first step of the proof of Lemma 4.25 which we summarize as follows.

Corollary 4.30.
For all $\mathbf{u} \in \mathcal{B}$ there exists a sequence $(u_n)_{n\in\mathbb{N}}$, $u_n \in \mathbf{A}_1^*$, such that against any $(\tau_n)_{n\in\mathbb{N}}$, $\tau_n \in \Sigma_2^p$, there exists $\mathbf{X} \subseteq \mathcal{M}$ with $\mathbf{X} \cap \mathbf{u} \neq \emptyset$ that is realized by $((u_n, \tau_n))_{n\in\mathbb{N}}$ with height $h = 9|\mathcal{M}|^2|\mathcal{B}|$.

To summarize, for all elements $\mathbf{u}$ of the belief monoid, we can find some sequence of finite words over the alphabet $\mathbf{A}_1$, $(u_n)_{n\in\mathbb{N}}$ such that against all the replies of Min $(\tau_n)_{n\in\mathbb{N}}$, we can construct k-decomposition trees of (u_n, τ_n) for larger and larger k and n, whose type is identical and equal to some element U of $\mathbf{u}$.

Now in the second step of the proof of Lemma 4.25, we will use these k-decomposition trees to prove bounds for $\mathbb{P}_s^{u_n,\tau_n}(t)$, $s, t \in \mathbf{S_1}$, such that asymptotically it behaves like U.

We will start with a lower bound. The same proof can be found for probabilistic automata in [Fijalkow *et al.*, 2012].

In the lemmas above, we have used the non-extended version of the Markov and belief monoids, for the sake of clarity, even though they all hold for the extended versions of the monoids as well.

Lemma 4.31.
Assume that the half-blind game is leaktight. There exists a function $L : \mathbb{N}^2 \to \mathbb{R}^{>0}$, such that for all words $p = (w, \tau) \in \mathfrak{A}^*$, T a k-decomposition tree of p of height at most h with type $(W, \widetilde{W}) \in \widetilde{\mathcal{M}}$, and $s, t \in \mathbf{S_1}$,

$$W(s,t) = 1 \implies \mathbb{P}_s^{w,\tau}(t) \geq L(h,k), \text{ and} \tag{55}$$

$$\widetilde{W}(s,t) = 1 \iff \mathbb{P}_s^{w,\tau}(t) > 0 \tag{56}$$

In other words, for any k-decomposition tree of a finite play p, the type $(W, \widetilde{W})$ of the tree gives the behavior of the game under p, in the way that if $W(s,t) = 1$ then we can bound from below the probability to go from s to t, furthermore this probability is nonzero if and only if $\widetilde{W}(s,t) = 1$.

Proof.
We start from the leaves, and walk up the tree, while proving the lower bound for all the nodes of the tree.

Leaves. The leaves have the label $(a, (B_{a,\alpha}, B_{a,\alpha}))$ for some $a \in \mathbf{A_1}$ and $\alpha \in \mathbf{S_2} \to \mathbf{A_2}$. The statements (55) and (56) for (a, α) and $(B_{a,\alpha})$ follow immediately from the definition of $B_{a,\alpha}$, for the bound m, the smallest nonzero probability transition that appears in the game.

Product nodes. Assume that there is a lower bound L for the children of the product node, that are labeled $(u, (U, \widetilde{U}))$ and $(v, (V, \widetilde{V}))$. Then it is possible to prove the lower bound L^2 for the parent node that is labeled by $(uv, \widetilde{U}\widetilde{V})$.

Idempotent nodes. Similarly to the product nodes above, if there is a lower bound L for the children, then the parent has the bound L^k, because the number of children is less than k.

Iteration nodes. Assume that the lemma holds for the n children, $n \geq k$, that are labeled by

$$(u_1, (U, \widetilde{U})), (u_2, (U, \widetilde{U})), \ldots, (u_n, (U, \widetilde{U})),$$

for the lower bound L, where $(U, \widetilde{U})$ is idempotent. From this assumption, (56) follows easily, so we prove (55). Let $s, t \in \mathbf{S_1}$ be such that $U^\#(s,t) = 1$. Then

$$\mathbb{P}_s^{u_1 \cdots u_n}(t) \geq \mathbb{P}_s^{u_1}(t) \sum_{s' \in \mathbf{S_1}} \mathbb{P}_t^{u_2 \cdots u_{n-1}}(s') \mathbb{P}_{s'}^{u_n}(t). \tag{57}$$

The state t is U-recurrent, this is because we have assumed $U^\#(s,t) = 1$. Moreover since $\widetilde{\mathcal{M}}$ is leaktight, the element (U, wU) is not a leak, therefore for all $s' \in \mathbf{S_1}$, $\mathbb{P}_t^{u_2 \cdots u_{n-1}}(s') > 0$, implies that $\widetilde{U}(t,s') = 1$, from where we have

$U(s',t) = 1$. Now from the induction hypothesis (that the lemma holds for the children) we conclude that $\mathbb{P}_{s'}^{u_n}(t) \geq L$. This together with (57) implies that

$$\mathbb{P}_{s}^{u_1 \cdots u_n}(t) \geq L^2.$$

From the four cases above, it follows that the lemma holds for the root node for the bound $L(h,k) = m^{k^h}$, where m is the smallest nonzero probability transition of the game. ∎

Now we are going to prove a dual lemma to the above, one that shows an upper bound that vanishes as k tends to infinity. This will conclude the second step and Lemma 4.25 itself.

Set

$$L = L(9|\widetilde{\mathcal{M}}|^2, 2),$$

where $L : \mathbb{N}^2 \to \mathbb{R}^{>0}$ is as above.

Lemma 4.32.
Assume that $\widetilde{\mathcal{M}}$ is leaktight. Let $h \in \mathbb{N}$, and set $K_h \in \mathbb{N}$ such that $h2^h(1 - L)^{K_h} < L$ and $K_h > |\mathbf{S_1}|$.

For all $p = (w, \tau) \in \mathfrak{A}^*$, $k > K_h$ and T a k-decomposition tree of p of height at most h with the root node labeled by $(p, (W, \widetilde{W}))$, for all $s, t \in \mathbf{S_1}$

$$W(s,t) = 0 \implies \mathbb{P}_{s}^{w,\tau}(t) \leq h2^h(1 - L^{|\mathbf{S_1}|})^{\lfloor k/|\mathbf{S_1}| \rfloor}.$$

Proof.
We walk up the tree, starting from the leaves up to the root node, while proving some upper bounds that are always smaller than $h2^h(1 - L^{|\mathbf{S_1}|})^{\lfloor k/|\mathbf{S_1}| \rfloor}$.

Leaves. The leaves have an upper bound of 0, this follows from the definitions.

Product nodes. Assume that the lemma holds for the bound F for the children labeled $(u, (U, \widetilde{U}))$ and $(v, (V, \widetilde{V}))$. Let $s, t \in \mathbf{S_1}$ be such that $UV(s,t) = 0$. It follows that

$$\mathbb{P}_{s}^{uv}(t) \leq 2F.$$

Idempotent nodes. Assume that we have the children whose label's left components are $p_1, \ldots, p_j, j < k$ and each label's right component is the same idempotent $(W, \widetilde{W})$. Let $s, t \in \mathbf{S_1}$ such that $W(s,t) = 0$. By the induction hypothesis the upper bound F holds for all the children.

Denote by ρ the set of all paths $s_0 s_1 \cdots s_j$ such that $s_0 = s$, $s_j = t$ and $\widetilde{W}(s_i, s_{i+1}) = 1$ for all $0 \leq i \leq j-1$. Since $W(s,t) = 0$ for all $\pi = s_0 \cdots s_j \in$

ρ there exists $0 \leq C(\pi) \leq j-1$ such that $W(s_{C(\pi)}, s_{C(\pi)+1}) = 0$ and for all $0 \leq i \leq C(\pi)-1$, $W(s_i, s_{i+1}) = 1$. Define ρ' to be the set of such prefixes, i.e.

$$\rho' = \{s_0 \cdots s_{C(\pi)} \mid \pi = s_0 \cdots s_j \in \rho\}.$$

The set ρ' is nonempty because there exists some $r \in \mathbf{S_1}$ such that $W(s,r) = 1$.

Then we have

$$\begin{aligned}
\mathbb{P}_s^{p_1 \cdots p_j}(t) &= \sum_{s_0 \cdots s_j \in \rho} \mathbb{P}_{s_0}^{p_1}(s_1) \cdots \mathbb{P}_{s_{j-1}}^{p_j}(s_j) \\
&\leq \sum_{\pi = s_0 \cdots s_{C(\pi)} \in \rho'} \mathbb{P}_{s_0}^{p_1} \cdots \mathbb{P}_{s_{C(\pi)-1}}^{p_{C(\pi)}}(s_{C(\pi)}) \cdot F \\
&\leq 2F,
\end{aligned}$$

Iteration nodes. Assume that we have the children whose label's left components are $p_1, \ldots, p_j$, $j \geq k$ and each label's right component is the same idempotent $(W, \widetilde{W})$ and for whom the upper bound F holds. Let $s, t \in \mathbf{S_1}$ be such that $W^{\#}(s,t) = 0$. In case $W(s,t) = 0$ a proof like the one above for idempotent nodes gives $2F$ as the upper bound. Therefore we assume that $W(s,t) = 1$. Then by definition t is W-transient and it communicates with some recurrence classes whose union we denote by $S_r \subseteq \mathbf{S_1}$. We will prove that for all $0 \leq i < j' \leq j$ such that $j' - i \geq |\mathbf{S_1}|$ there exists $i \leq i' \leq j'$ such that $i' - i \leq |\mathbf{S_1}|$ and

$$\mathbb{P}_t^{p_i \cdots p_{i'}}(S_r) \geq L^{|\mathbf{S_1}|}. \tag{58}$$

Let $i \in \{0, \ldots j\}$, then there exists a 2-decomposition tree T_i for the word p_i, with type $(W_i, \widetilde{W})$. It is possible that $W \neq W_i$, but for all $s', t' \in \mathbf{S_1}$, $W(s', t') = 0$ implies that $W_i(s', t') = 0$. This is because by the induction hypothesis, if $W(s', t') = 0$, we know that $\mathbb{P}_{s'}^{u_i}(t') \leq F$ whereas according to Lemma 4.31 for T_i, if $W_i(s', t') = 1$ we have $\mathbb{P}_{s'}^{u_i}(t') \geq L$, from $F \leq h2^h \cdot (1-L)^k$ and our choice of k, superior to K_h, this is a contradiction, hence $W_i(s', t') = 0$.

Let S_t be the set of states that are W-reachable from t (for all $t' \in S_t$, $W(t, t') = 1$) but not in S_r. These states are all W-transient and moreover for all $i \in \{0, \ldots, j\}$, there exists a W_i path from t, and any state in S_t, to some element in S_r. This is because for all $t' \in S_t \cup \{t\}, r \in S_r$, $\widetilde{W}(t', r) = 1$, and there is no W_i path from S_r to $S_t \cup \{t\}$, if there was no W_i path from $S_t \cup \{t\}$ we could construct a leak, which contradicts the hypothesis that $\widetilde{\mathcal{M}}$ is leaktight. Similarly, for $0 \leq i < j' \leq j$ such that $i - j' \geq |\mathbf{S_1}|$, if $W_i \cdots W_{j'}(t, S_r) = 0$ we can construct a leak by repeating a factor of $W_i \cdots W_{j'}$, hence we can assume that there exists $i \leq i' \leq j'$, such that $i' - i \leq |\mathbf{S_1}|$ and $W_i \cdots W_{i'}(t, S_r) = 1$. Then it follows from Lemma 4.31 that $\mathbb{P}_t^{p_i \cdots p_{i'}}(S_r) \geq L^{|\mathbf{S_1}|}$ which concludes (58).

Let ρ be the set of all paths $s_0 \cdots s_j$ such that $s_0 = s, s_j = t$ and $\widetilde{W}(s_i, s_{i+1}) = 1$ for all $0 \leq i \leq j-1$. We partition ρ into the set ρ_1 of all the paths that pass through S_r and ρ_2 the set of all paths that do not. Since t is W-transient, for all $r \in S_r$, $W(r,t) = 0$, consequently we can use the argument above for the idempotent nodes to give $2F$ as an upper bound for the probability of the event that constitutes the union of all the sets in ρ_1. As for ρ_2, because of transience of t and (58) the probability of the union of all the paths in ρ_2 can be bounded above by $2(1 - L^{\mathbf{S_1}})^{\lfloor j/|\mathbf{S_1}| \rfloor}$.

Walking up the tree of height h for the upper bounds above, we conclude the lemma for the root node. ∎

Lemma 4.25 is a consequence of Corollary 4.30 and Lemma 4.32.

4.6 COMPLETENESS

We shall prove the second part of the correctness of the belief monoid algorithm in this section: if the belief monoid of a leaktight half-blind game has no maxmin reachability witness then $\underline{val}(s) < 1$ where s is the initial state of the game.

Theorem 4.33 (Completeness).
If the belief monoid of a half-blind leaktight game has no maxmin reachability witness then

$$\underline{val}(s) < 1,$$

where $s \in \mathbf{S_1}$ is the initial state of the game.

For the soundness we were justifying the elements of $\mathbf{u} \in \mathcal{B}$, by demonstrating that for all of them there is some word over $\mathbf{A}_1$ that realizes them, *i.e.* for all replies of Min, the behavior is described by some $U \in \mathbf{u}$. In this section we will justify the $U \in \mathbf{u}$, by proving that there exists some strategy of Min such that the behavior is described by U. In other words, we are not abstracting more behaviors than possible.

Let us be more precise, and summarize that intuitive idea into the following definition.

Definition 4.34 (μ-faithful abstraction).
Let $u \in \mathbf{A}_1^*$, and $\mu > 0$. We say that $\mathbf{u} \in \widetilde{\mathcal{B}}$ is a μ-faithful abstraction of u if for all $(U, \widetilde{U}) \in \mathbf{u}$ there exists some $\tau \in \Sigma_2^p$ such that for all $s, t \in \mathbf{S_1}$,

$$\widetilde{U}(s,t) = 1 \iff \mathbb{P}_s^{u,\tau}(t) > 0, \tag{59}$$

$$U(s,t) = 1 \implies \mathbb{P}_s^{u,\tau}(t) \geq \mu. \tag{60}$$

In the definition above μ is the lower bound. We want to prove that there exists some $\mu > 0$, such that *all* words have some element of the belief monoid that is a μ-faithful abstraction of them. This is sufficient for Theorem 4.33 for the following reason. If $\underline{val}(s) = 1$, then there exists some $u \in \mathbf{A}_1^*$ such that against all the strategies of Min the chance of reaching the set of final states is at least $1 - \mu^2$. But for the word u there exists an element of the belief monoid $\mathbf{u}$ (that is not a maxmin reachability witness) that is a μ-faithful abstraction of u. Since $\mathbf{u}$ is not a maxmin reachability witness there is some $U \in \mathbf{u}$ such that $U(s,t) = 1$ where s is the initial state and $t \notin \mathbf{F}$. Therefore there exists some $\tau \in \Sigma_2^p$ such that $\mathbb{P}_s^{u,\tau}(t) \leq 1 - \mu$, a contradiction, hence $\underline{val}(s) < 1$. As a consequence we concentrate our efforts into the proof of the following lemma.

Lemma 4.35.
There exists $\mu > 0$ such that all $u \in \mathbf{A}_1^*$ have some $\mathbf{u} \in \widetilde{\mathcal{B}}$ that is their μ-faithful abstraction.

First observe that for letters $a \in \mathbf{A}_1$, $\mathbf{a} \in \widetilde{\mathcal{B}}$ is a m-faithful abstraction, where $m > 0$ is the smallest nonzero probability transition that appears in the game. Moreover we can concatenate two words, but unfortunately the lower bound diminishes:

Lemma 4.36.
Let $\mathbf{u}, \mathbf{v} \in \widetilde{\mathcal{B}}$ be μ-faithful abstraction of $u \in \mathbf{A}_1^*$ and $v \in \mathbf{A}_1^*$ respectively, then $\mathbf{uv}$ is a μ^2-faithful abstraction of uv.

We can use this lemma repeatedly, but then the lower bound depends on the length of the word, and diminishes with it. This is because by playing longer

and longer words Max can make the probability to go to the transient states arbitrarily small, so the lower bound gets arbitrarily small as well. So we need to abstract these longer words by the iterated elements of the monoid.

The idea is to use k-decomposition trees for the alphabet $\mathbf{A}_1$ and the monoid $\widetilde{\mathcal{B}}$. Given some (longer) word u, we construct its k-decomposition tree and then prove that its type is a μ-faithful abstraction of u, and moreover that μ depends only on the height of the tree and not the length of the word, at which point, Lemma 4.17 comes in. The leaktight hypothesis is used to show that for iteration nodes the bound does not decrease as a function of the number of children.

Let $\mathcal{N} = 2^{9|\widetilde{\mathcal{M}}|^2}$. The essential part of this section is the proof of the following lemma.

Lemma 4.37.
Let $u \in \Sigma_1$, such that $u = u_1 \cdots u_n$, where $n > \mathcal{N} = 2^{9|\widetilde{\mathcal{M}}|}$, and $\mathbf{u} \in \widetilde{\mathcal{B}}$ and idempotent element that is a μ-faithful abstraction of u_i, $1 \leq i \leq n$, for some $\mu > 0$. If $\mathbf{u}$ is not a leak then $\mathbf{u}^\#$ is a $\mu^{\mathcal{N}+1}$-faithful abstraction of u.

We argue why this lemma is sufficient for Lemma 4.35. Given a word $u \in \mathbf{A}_1^*$, we construct a $\mathcal{N}$-decomposition tree of u of height at most $9|\widetilde{\mathcal{B}}|^2$. Such a tree always exists thanks to Lemma 4.17. For the leaves we can prove the bound $m > 0$, then we can propagate it up the tree as follows, if the bound μ holds for the children, in case of a product node we can prove μ^2, in case of an idempotent node we can prove $\mu^{\mathcal{N}}$ and in case of an iteration node $\mu^{\mathcal{N}+1}$. Since the height of the tree is at most $h = 9|\widetilde{\mathcal{B}}|^2$ we have the lower bound

$$\mu = m^{h(\mathcal{N}+1)},$$

that holds for all $u \in \mathbf{A}_1^*$.

In the proof that follows we will use the following notation, for some $B \in \mathcal{M}$ and $s, t \in \mathbf{S_1}$, we write $s \xrightarrow{B} t$ if and only if $B(s,t) = 1$, otherwise we write $s \not\xrightarrow{B} t$.

Proof.
Let $(W, \widetilde{W}) \in \mathbf{u}^\#$ we will construct a strategy $\tau \in \Sigma_2^p$ such that (59) and (60) in Definition 4.34 hold, for $(W, \widetilde{W})$ the word u and the bound $\mu' = \mu^{\mathcal{N}+1}$. Let us first assume that $(W, \widetilde{W})$ is such that

$$W = F_1 G_1^\# \cdots F_k G_k^\# F_{k+1},$$
$$\widetilde{W} = \widetilde{F}_1 \widetilde{G}_1 \cdots \widetilde{F}_k \widetilde{G}_k \widetilde{F}_{k+1},$$

where $(F_i, \widetilde{F}_i) \in \mathbf{u}$,$(G_i, \widetilde{G}_i) \in \mathbf{u}$ and $(G_i, \widetilde{G}_i)$ are idempotent.

The set of #-expressions of $\mathbf{u} \subseteq \mathcal{M}$ denoted by $\mathcal{E}(\mathbf{u})$ is a language defined by the grammar:

$$\mathcal{E}(\mathbf{u}) := \mathbf{u} \mid \mathcal{E}(\mathbf{u}) \cdot \mathcal{E}(\mathbf{u}) \mid (\mathcal{E}(\mathbf{u}))^{\#},$$

so the terminal symbols are the elements of $\mathbf{u}$. There is $\gamma_{\mathbf{u}}$, a natural function mapping $\mathcal{E}(\mathbf{u})$ to $\mathcal{M}$, i.e. the function that is the identity when restricted to the terminal symbols, otherwise $\gamma_{\mathbf{u}}(e \cdot e') = \gamma_{\mathbf{u}}(e)\gamma_{\mathbf{u}}(e')$, and $\gamma_{\mathbf{u}}(e^{\#}) = (\gamma_{\mathbf{u}}(e))^{\#}$. Given $e \in \mathcal{E}(\mathbf{u})$ we define its #-*height* as the number of the deepest nesting of #. E.g. $\# - height(U^{\#}(VW^{\#})^{\#}) = 2$.

We can safely make the assumption on the form of $(W, \widetilde{W})$ above because for all $(U, \widetilde{U}) \in \mathbf{u}$ we can find a #-expression $e \in \mathcal{E}(\mathbf{u})$ whose #-height is 1, such that $\gamma_{\mathbf{u}}(e) = (U', \widetilde{U})$ and for all $s, t \in \mathbf{S_1}$ $U(s,t) = 1 \implies U'(s,t) = 1$. This is because when we iterate we are removing edges.

Since $\mathbf{u}$ is a μ-faithful abstraction of u_i,$1 \leq i \leq n$, for all $(U, \widetilde{U})$ in $\mathbf{u}$ there is a strategy in Σ_2^p such that (59) and (60) hold. Let τ_1 be such a strategy for $(F_1, \widetilde{F}_1)$, τ_2 for $(G_1, \widetilde{G}_1)$ and so on until τ_{2k+1} for the element $(F_{k+1}, \widetilde{F}_{k+1})$. We define the strategy τ by assigning one of the τ_i to some part of the word in the following way:

- against u_1 play τ_1,
- against u_2 play τ_2, play τ_2 also against $u_3, u_4, \ldots, u_{n-2k+1}$ each,
- against u_{n-2k+2} play τ_3, etc., in general against $u_{n-2k+1+i}$ play τ_{i+2}, $1 \leq i \leq 2k-1$.

One can visualize this in the following way.

$$\tau := \begin{array}{c|c|c|c|c|c|c} u_1 & u_2, u_3, \ldots, u_{n-2k+1} & u_{n-2k+2} & \cdots & u_{n-2k+2k-1} & u_n \\ \tau_1 & \tau_2 & \tau_3 & & \tau_{2k} & \tau_{2k+1} \\ F_1 & G_1 & F_2 & & G_k & F_{k+1} \end{array}.$$

This means that τ plays according to τ_2 against u_2 then it keeps playing according to τ_2 against u_3 and so on until u_{n-2k+1} is read. Note that it is well defined since we have assumed that $n > \mathcal{N}$.

Now we prove (59) for τ and u.

($\implies$) Let $s, t \in \mathbf{S_1}$ be such that $s \xrightarrow{\widetilde{W}} t$. Since $\widetilde{W} = \widetilde{F}_1\widetilde{G}_1 \cdots \widetilde{G}_k\widetilde{F}_{k+1}$ and $\widetilde{G}_1$ is idempotent there exist $s_1, \ldots, s_{n-1} \in \mathbf{S_1}$ such that

$$s \xrightarrow{\widetilde{F}_1} s_1 \xrightarrow{\widetilde{G}_1} \cdots \xrightarrow{\widetilde{G}_1} s_{n-2k} \xrightarrow{\widetilde{F}_2} s_{n-2k+1} \xrightarrow{\widetilde{G}_2} s_{n-2k+2} \cdots \xrightarrow{\widetilde{G}_k} s_{n-1} \xrightarrow{\widetilde{F}_{k+1}} t. \quad (61)$$

Let $F(s_1, \ldots, s_{n-1})$ be equal to

$$\mathbb{P}_s^{u_1,\tau_1}(s_1)\mathbb{P}_{s_1}^{u_2,\tau_2}(s_2) \cdots \mathbb{P}_{s_{n-2k-1}}^{u_{n-2k+1},\tau_2}(s_{n-2k})\mathbb{P}_{s_{n-2k}}^{u_{n-2k+2},\tau_3}(s_{n-2k+1}) \cdots \mathbb{P}_{s_{n-1}}^{u_n,\tau_{2k+1}}(t).$$

Then by the choice of τ we have $\mathbb{P}_s^{u,\tau}(t) \geq F(s_1, \ldots, s_{n-1})$. Since $\mathbf{u}$ is a μ-faithful abstraction of u_i, (61) implies that every factor of $F(s_1, \ldots, s_{n-1})$ is positive, hence $\mathbb{P}_s^{u,\tau}(t) > 0$.

($\Longleftarrow$) Let $s, t \in \mathbf{S_1}$ be such that $\mathbb{P}_s^{u,\tau}(t) > 0$, then similarly to above there must exist states $s_1, \ldots, s_{n-1}$ such that

$$\mathbb{P}_s^{u,\tau}(t) \geq F(s_1, \ldots, s_{n-1}) > 0.$$

This implies (61), since $\mathbf{u}$ is a μ-faithful abstraction of all u_i, and in turn, (61) implies that $s \xrightarrow{\widetilde{W}} t$, since $\widetilde{W} = \widetilde{F}_1 \widetilde{G}_1 \cdots \widetilde{G}_k \widetilde{F}_{k+1}$.

Now we prove (60) for τ and u and the bound $\mu' = \mu^{\mathcal{N}+1}$. Let $s, t \in \mathbf{S_1}$ such that $s \xrightarrow{W} t$. Then there exists states $s_1, \ldots, s_{2k}$ such that

$$s \xrightarrow{F_1} s_1 \xrightarrow{G_1^\#} s_2 \xrightarrow{F_2} \cdots \xrightarrow{G_k^\#} s_{2k} \xrightarrow{F_{k+1}} t. \tag{62}$$

First we will show that

$$\mathbb{P}_{s_1}^{u_2,\ldots,u_{n-2k+1},\tau'}(s_2) \geq \mu^2, \tag{63}$$

where τ' is the strategy that plays τ_2 against u_2, and against u_3 and so on. This is exactly what the strategy τ does, after it reads u_1. Then we have

$$\mathbb{P}_{s_1}^{u_2 \cdots u_{n-2k+1},\tau'}(s_2) \geq \mathbb{P}_{s_1}^{u_2,\tau_2}(s_2) \sum_{s' \in \mathbf{S_1}} \mathbb{P}_{s_2}^{u_3 \cdots u_{n-2k},\tau''}(s') \mathbb{P}_{s'}^{u_{n-2k+1},\tau_2}(s_2),$$

where τ'' is the strategy that plays τ_2 against u_3 ,and against u_4 and so on. The strategy τ'' is the same as τ' just shifted by the first part u_2. From (62) $s_1 \xrightarrow{G_1^\#} s_2$, which implies that s_2 is G_1-recurrent, $s_1 \xrightarrow{\widetilde{G}_1} s_2$ and $s_1 \xrightarrow{G_1} s_2$. By the choice of τ_2, because $s_1 \xrightarrow{G_1} s_2$, we have

$$\mathbb{P}_{s_1}^{u_2 \cdots u_{n-2k+1},\tau'}(s_2) \geq \mu \sum_{s' \in \mathbf{S_1}} \mathbb{P}_{s_2}^{u_3 \cdots u_{n-2k},\tau''}(s') \mathbb{P}_{s'}^{u_{n-2k+1},\tau_2}(s_2). \tag{64}$$

Let s' be such that $\mathbb{P}_{s_2}^{u_3 \cdots u_{n-2k},\tau''}(s') > 0$. Then from the definition of τ'', $s_2 \xrightarrow{\widetilde{G}_1^{n-2k-3}} s'$ and since $\widetilde{G}_1$ is idempotent $s_2 \xrightarrow{\widetilde{G}_1} s'$. We will prove that $s' \xrightarrow{G_1} s_2$. There are two cases:

- s' is G_1-recurrent: then both s' and s_2 are G_1-recurrent, and $s_2 \xrightarrow{\widetilde{G}_1} s'$. Since we have assumed that $\mathbf{u}$ is not a leak, then $s' \xrightarrow{G_1} s_2$.
- s' is G_1-transient: there exists some state r that is G_1-recurrent, such that $s' \xrightarrow{G_1} r$ and $r \not\xrightarrow{G_1} s'$. Now $s' \xrightarrow{G_1} r$ implies that $s' \xrightarrow{\widetilde{G}_1} r$, and from idempotency of $\widetilde{G}_1$, $s_2 \xrightarrow{\widetilde{G}_1} r$. Then from the argument for the case above $r \xrightarrow{G_1} s_2$, and finally from idempotency of G_1, $s' \xrightarrow{G_1} s_2$.

We have shown that for all s' such that $\mathbb{P}_{s_2}^{u_3\cdots u_{n-2k},\tau''}(s') > 0$, $s' \xrightarrow{G_1} s_2$. As a consequence, from the choice of τ_2 and (64) we have

$$\mathbb{P}_{s_1}^{u_2\cdots u_{n-2k+1},\tau'}(s_2) \geq \mu^2.$$

To finish up with the proof of (60), for all $s, s' \in \mathbf{S_1}$ and G_i, $s \xrightarrow{G_i^\#} s'$ implies that $s \xrightarrow{G_i} s'$, therefore from (62) we have

$$s \xrightarrow{F_1} s_1 \xrightarrow{G_1^\#} s_2 \xrightarrow{F_2} s_3 \xrightarrow{G_2} \cdots \xrightarrow{G_k} s_{2k} \xrightarrow{F_{k+1}} t, \tag{65}$$

so for all G_i, $2 \leq i \leq k$, we write G_i instead of $G_i^\#$. Then by the choice of the strategies τ_i and the definition of τ,

$$\mathbb{P}_s^{u,\tau}(t) \geq \mathbb{P}_s^{u_1,\tau_1}(s_1)\mathbb{P}_{s_1}^{u_2,\ldots,u_{n-2k+1},\tau'}(s_2)\cdots\mathbb{P}_{s_{2k}}^{u_n,\tau_{2k+1}}(t) \geq \mu\cdot\mu^2\cdot\mu^{2k-1} = \mu^{2k+2},$$

where for the last inequality we have used (63) and (65). Since $2k+1 \leq \mathcal{N}$, this concludes the proof of (60) for τ,u and the bound $\mu' = \mu^{\mathcal{N}+1}$. ∎

4.7 POWER OF STRATEGIES AND EXAMPLES

A natural question when studying any type of game is how powerful should the strategies of the players be. In this section we are going to discuss whether randomization, or infinite memory allows the players to win strictly more.

The maxmin reachability question asks whether the quantity

$$\underline{val}(s) = \sup_{w\in\Sigma_1}\inf_{\tau\in\Sigma_2}\mathbb{P}_s^{w,\tau}(\mathbf{F}),$$

is equal to 1. That is, whether for all $\epsilon > 0$ there exists some finite word w such that against all (behavioral) strategies τ of Min the chance of reaching the set of final states from the initial state s is at least $1-\epsilon$.

This question is motivated from considering the model of half-blind games which serve as a generalization of probabilistic finite automata to two player games with a perfectly-informed adversary. These games can be seen as probabilistic automata whose probabilistic transitions we do not know exactly, because they are controlled by the environment. Another reason for considering this model is the notion of leaks. It is interesting to explore where else can it be used except for probabilistic automata, and half-blind games are a prime candidate.

Nevertheless other quantities are just as interesting to study. Primarily

$$val(s) = \sup_{w\in\Sigma_1^m}\inf_{\tau\in\Sigma_2}\mathbb{P}_s^{w,\tau}(\mathbf{F}),$$

where $\Sigma_1^m = \Delta(\mathbf{A}_1^*)$ is the set of *mixed* words. This is induces the question of whether for all $\epsilon > 0$ there exists a *mixed* strategy for Max (*i.e.* a distribution over the set of finite words) such that against all the strategies of Min, the set of final states is reached with probability at least $1 - \epsilon$. This problem is undecidable as well, for the same reason as the maxmin reachability problem, that is, its decidability implies the decidability of the value 1 problem for probabilistic finite automata. Observe that in general

$$\underline{val}(s) \leq val(s),$$

and that if $\underline{val} < val(s) = 1$ then Max needs to mix between larger and larger sets of finite words in order to reach the set of final states with a larger and larger probability. An important argument for the study of the quantity $val(s)$ is that under these two types of strategies: mixed finite words for Max and behavioral strategies for Min the game admits a value:

Theorem 4.38 ([Gimbert *et al.*, 2016]).

$$val(s) = \sup_{w \in \Sigma_1^m} \inf_{\tau \in \Sigma_2} \mathbb{P}_s^{w,\tau}(\mathbf{F}) = \inf_{\tau \in \Sigma_2} \sup_{w \in \Sigma_1^m} \mathbb{P}_s^{w,\tau}(\mathbf{F}).$$

A natural question is whether we can relate $\underline{val}(s)$ and $val(s)$, particularly whether there exists some game where $\underline{val}(s) < val(s)$. Indeed there is, we have shown this in Example 13. For this example

$$\underline{val}(s) = \sup_{w \in \Sigma_1} \inf_{\tau \in \Sigma_2} \mathbb{P}_i^{w,\tau}(\{f\}) < \inf_{\tau \in \Sigma_2} \sup_{w \in \Sigma_1} \mathbb{P}_i^{w,\tau}(\{f\}).$$

But the quantity on the right hand side is equal to $val(s)$, this is because once the strategy of Min has been fixed, Max cannot gain anything by randomizing his choice of words, this is helpful to him only if he has to choose his strategy before Min chooses his. Its purpose is to hide information (the exact strategy) from Min.

Unfortunately the belief monoid algorithm does not seem to tell us anything about $val(s)$, except for a lower bound, moreover the analysis that we have done in this chapter is not likely to be helpful to resolve the problem of whether $val(s) = 1$.

In other words, we are able to treat the $\sup_{w \in \Sigma_1} \inf_{\tau \in \Sigma_2} \mathbb{P}_s^{w,\tau}(\mathbf{F})$ but not $\inf_{\tau \in \Sigma_2} \sup_{w \in \Sigma_1} \mathbb{P}_s^{w,\tau}(\mathbf{F})$. The essential difficulty is that strategies are complex objects, in the former case, that of $\sup_{w \in \Sigma_1} \inf_{\tau \in \Sigma_2} \mathbb{P}_s^{w,\tau}(\mathbf{F})$, we need only analyze a simpler set of strategies for Min thanks to Lemma 4.6. But when Min fixes

its strategy first we cannot *a priori* make such a simplification. Nevertheless this raises an interesting question: does Min need infinite memory strategies?

Denote by Σ_2^f the set of finite-memory strategies for Min, these are finite state transducers that read finite histories and output elements of $\Delta(\mathbf{A}_2)$. Obviously

$$val^f(s) = \inf_{\tau \in \Sigma_2^f} \sup_{w \in \Sigma_1} \geq val(s),$$

but does there exist an example where Min stands to gain more by playing with an infinite memory strategy? At first sight this seems unlikely when one considers that Min is much more advantaged, being perfectly informed while his adversary has zero information. But it turns out that in order for Min to profit maximally from this advantage he might need infinite memory strategies. We are going to prove this in the remainder of this section.

Proposition 4.39.
There exists a half-blind game with initial state s, for which

$$val^f(s) > val(s).$$

The proof of Proposition 4.39 will be based on the following gadget.

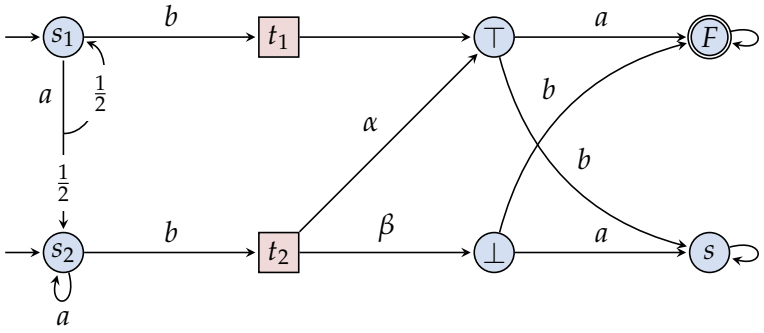


Figure 21: A gadget

We give the main idea behind the gadget. Max wants to be able to know whether he is in state top or bottom after playing his first b so that he can go to the final state. The objective of Min is to make the probability of being in the top state equal to that of being in the bottom state, so that Max cannot win more than $\frac{1}{2}$. In order to do this, when it is his turn to make the choice between α and β (or a mixing of them) he has to know the exact probability

distribution over t_1 and t_2. But this is impossible to keep track with a finite memory strategy, because Max plays too many a's for Min's small memory. Hence Max can always win slightly more than $\frac{1}{2}$. We then use this gadget in a new game that emphasizes the importance of these winnings and prove that in that game $\frac{1}{2} = val(s) < val^f(s) = 1$. Let us now demonstrate these ideas formally.

The game starts either at state s_1 or s_2 with equal probability. Max can play a series of a's but eventually has to play a b if he wants to make progress. After this, Min observes whether the game is in the state t_1 or t_2. In case it is in t_1, Min has no choice and proceeds to state $\top$. In case it is in t_2 Min can choose between α and β to go either to state $\top$ or to state $\bot$. Then Max has to guess which one it is. If the guess is right he wins, if it is wrong he loses by going to the sink state. The goal of Min is to keep track of the probability distribution on the states of the game such that when it is his time to make a decision, he plays a mixed action (between α and β) such that the probability to be in $\top$ is equal to the probability to be in $\bot$ equal to $\frac{1}{2}$. Keeping track of the distribution will be impossible with a finite memory strategy because the sequence of a's that Max plays can be arbitrarily long.

Observe that $val(\gamma) = \frac{1}{2}$, where γ is the initial distribution, *i.e.* $\gamma(s_1) = \gamma(s_2) = \frac{1}{2}$, by giving the optimal strategies as follows. Max can mix the two words ba and bb with equal probability. Call this mixed word w. Then for all strategies τ for Min, we have $\mathbb{P}_\gamma^{w,\tau}(F) = \frac{1}{2}$. In the other hand, after a b is played, the probability to be in the state t_2 is always larger than $\frac{1}{2}$, $\mathbb{P}_\gamma^{a^n b,\tau}(t_2) \geq \frac{1}{2}$, and consequently Min has an optimal action such that both $\top$ and $\bot$ are reached with equal probability and equal to $\frac{1}{2}$. Moreover this optimal action can be played by Min, by keeping track of the distribution on t_1 and t_2 by counting the number of as that are played before b. Albeit this requires memory of unbounded size. We give a proof of this in what follows.

Assume that the game stops just before Min makes his action, then we have

$$\mathbb{P}_\gamma^{a^n b,\tau}(t_2) = 1 - \frac{1}{2} \cdot \frac{1}{2^n} = \frac{2^{n+1} - 1}{2^{n+1}}$$

Therefore if τ is optimal, after seeing $a^n b$ it would play the action β with the following probability,

$$\tau(a^n b)(\beta) = \frac{1}{2} \cdot \frac{2^{n+1}}{2^{n+1} - 1} = \frac{1}{2} \cdot \frac{1}{1 - \frac{1}{2^{n+1}}}.$$

With such a strategy, it is ensures that $\mathbb{P}_\gamma^{a^n b,\tau}(\top) = \mathbb{P}_\gamma^{a^n b,\tau}(\bot) = \frac{1}{2}$. We prove that this is impossible with a finite memory strategy.

The proof is by contradiction. Assume that the minimizer has a finite-memory strategy with m states such that against the word $a^n b$ it plays the action β

with probability $\frac{1}{2} \cdot \frac{1}{1-\frac{1}{2^{n+1}}}$. From the definition of a finite memory strategy, this implies that there exist two $m \times m$ stochastic matrices A and B, and $J \subset \{1, \ldots, m\}$ such that

$$\sum_{j \in J} (A^n B)_{i,j} = \frac{1}{2} \cdot \frac{1}{1 - \frac{1}{2^{n+1}}}, \tag{66}$$

where i is the initial memory location of the strategy, and for a matrix A we denote by $A_{i,j}$ the element on the ith row and jth column. We use the following well-known theorem. See *e.g.* [Gantmacher, 1959].

Theorem 4.40.
Let A be a square $m \times m$ stochastic matrix and $\lambda_1, \lambda_1, \ldots, \lambda_r$ ($r \leq m$) its distinct eigenvalues. Then for all $n > m$

$$(A^n)_{i,j} = \sum_{k=1}^{r} \lambda_k^n P_{ijk}(n),$$

where P_{ijk} are polynomials of smaller order than the multiplicity of λ_k.

Using Theorem 4.40 and doing a small calculation we see that there exist polynomials $P_1, P_2, \ldots, P_r$ such that for all $n > m$

$$\sum_{j \in J} (A^n B)_{i,j} = \sum_{k=1}^{r} \lambda_k^n P_k(n).$$

In the other hand the Taylor expansion for $\frac{1}{1-\frac{1}{2^{n+1}}}$ gives us

$$\sum_{j \in J} (A^n B)_{i,j} = \frac{1}{2} \cdot (1 + \frac{1}{2^{n+1}} + \frac{1}{2^{2(n+1)}} + \cdots).$$

Therefore

$$\sum_{k=1}^{r} \lambda_k^n P_k(n) = \frac{1}{2} \cdot (1 + \frac{1}{2^{n+1}} + \frac{1}{2^{2(n+1)}} + \cdots). \tag{67}$$

Now observe that for complex numbers $z_1, \ldots, z_m$, $m \geq 1$, with $|z_1| = |z_2| = \cdots = |z_m|$, real $c > 0$, and polynomials $f_1, \ldots, f_m$ on n of degree at most d,

$$\lim_{n \to \infty} \frac{c}{\sum_{i=1}^{m} z_i^n f_i(n)} = 1, \tag{68}$$

implies that $\sum_{i=1}^{m} z_i^n f_i(n) = \sum_{i=1}^{m} c_i z_i^n = c$ for some constants c_i. The reason being that (68) clearly cannot be true for $|z_i| < 1$, as for $|z_i| \geq 1$ assume that the dominating term of the denominator has the form $n^k \sum_{i=1}^{m} c_i z_i^n$ for constants c_i, then for (68) to hold we need $k = 0$. Hence $\sum_{i=1}^{m} z_i^n f_i(n) = \sum_{i=1}^{m} c_i z_i^n$, and similarly it is necessary that $|z_1| = |z_2| = \cdots = |z_m| = 1$. Finally, because of (68) we have $\sum_{i=1}^{m} c_i z_i^n = c$.

Assume without loss of generality that $|\lambda_1| = |\lambda_2| = \cdots = |\lambda_{r_1}|$, for some $1 \leq r_1 \leq r$ and that $|\lambda_1| \geq |\lambda_i|$, $1 \leq i \leq r$. The expression on the left hand side of (69) is dominated by $\sum_{k=1}^{r_1} \lambda_k^n P_k(n)$ whereas the expression on the right hand side is dominated by the leading term $\frac{1}{2}$.

Consequently, because of the equality above, it follows that

$$\lim_{n\to\infty} \frac{\frac{1}{2}}{\sum_{k=1}^{r_1} \lambda_k^n P_k(n)} = 1.$$

Applying (68) we have $\sum_{k=1}^{r_1} \lambda_k^n P_k(n) = \sum_{k=1}^{r_1} \lambda_k^n c_k = \frac{1}{2}$. We substract both of these equal quantities from (67), to get

$$\sum_{k=r_1}^{r} \lambda_k^n P_k(n) = \frac{1}{2} \cdot \left(\frac{1}{2^{n+1}} + \frac{1}{2^{2(n+1)}} + \cdots\right). \tag{69}$$

Repeating the same argument for the leading terms of (69) we have

$$\lim_{n\to\infty} \frac{\frac{1}{2^{n+2}}}{\sum_{k=r_1}^{r_2} \lambda_k^n P_k(n)} = \lim_{n\to\infty} \frac{1}{4\sum_{k=r_1}^{r_2} (2\lambda_k)^n P_k(n)} = 1.$$

Again, applying (68) we get $\sum_{k=r_1}^{r_2} (2\lambda_k)^n P_k(n) = \sum_{k=r_1}^{r_2} c_k' 2^n \lambda_k^n = 1/4$. Hence we can subtract the quantity $\frac{1}{2^{n+2}}$ from both sides in (69). Repeating the same argument for the eigenvalues that are left we conclude that

$$0 = \frac{1}{2} \cdot \left(\frac{1}{2^{r(n+1)}} + \frac{1}{2^{(r+1)(n+1)}} + \cdots\right),$$

which is clearly a contradiction therefore there are no two finite stochastic matrices A, B such that (66) holds, and consequently the minimizer has no finite-strategy that is optimal in achieving the $\frac{1}{2}$. Nevertheless for all $\epsilon > 0$ the minimizer has ϵ-optimal strategies that have finite memory. These strategies would constitute of counting the number of a's up to some length.

We have shown the following lemma.

Lemma 4.41.
In the game in Figure 21 for all finite memory strategies τ for the minimizer there exists a word w such that

$$\mathbb{P}_{\gamma}^{w,\tau}(F) > \frac{1}{2},$$

where γ is the initial distribution, $\gamma(s_1) = \gamma(s_2) = 1/2$.

In order to finish the proof of Proposition 4.39, now we give an example that gives a stronger property. We will use the game in Figure 21 in another game as a gadget. We then demonstrate that for this larger game it also holds that $val(i) = 1/2$ where i is the initial state but $val^f(i) = 1$, *i.e.* for all finite memory strategies τ and $\epsilon > 0$ there is a finite word that reaches the set of final states with probability larger than $1 - \epsilon$.

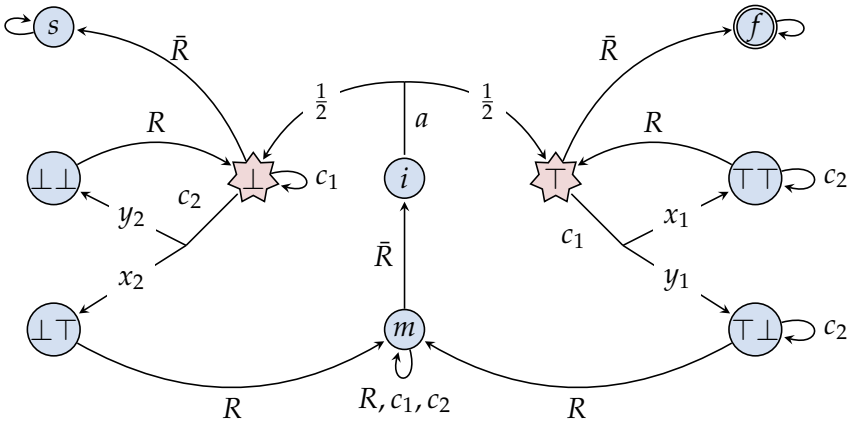


Figure 22: A game for which $val(i) = 1/2$

We give an informal description of the game in Figure 22. The state i is the initial state. A fair coin is tossed at i and if it results heads, then the game moves to state $\top$, otherwise it goes to state $\bot$. After this, we toss a biased coin in $\top$ by playing c_1, if we happen to be in $\bot$, playing c_1 would not change anything. At this point, another biased coin is tossed by playing c_2 as a result we are in one of the states $\bot\bot, \bot\top, \top\top, \top\bot$, after the two coin tosses. Repeating this process n times, by playing $a(c_1c_2R)^n$, we end up in state $\top$ if and only if we had $n+1$ heads and symmetrically we end up in state $\bot$ if and only if we have

tossed $n+1$ tails. Now we play $\bar{R}$, and by doing so we win if we have tossed $n+1$ consecutive heads, we lose if we have tossed $n+1$ consecutive tails and otherwise we go to the state i. If we repeat this process k times, by playing the word

$$(a(c_1c_2R)^n\bar{R})^k,$$

then the probability to win the game will be arbitrarily close to 1 (for well chosen n and k) if and only if the coin tosses are biased towards heads, that is $x_1, x_2 > 1/2$. The idea is to embed the gadget in Figure 21 in place of the states $\bot$ and $\top$.

For all k let

$$\mu_k = \mathbb{P}_i^{(a(c_1c_2R)^n\bar{R})^k}(\neg\{f,s\}),$$

the probability to be in any state except the sink (s) or final (f) state after the word $(a(c_1c_2R)^n\bar{R})^k$ has been played. Then we have

$$\begin{aligned}\mu_0 &= 1, \text{ and}\\ \mu_k &= \mu_{k-1}(1-\frac{1}{2}x_1^n-\frac{1}{2}y_2^n).\end{aligned}$$

Hence

$$\mu_k = (1-\frac{1}{2}x_1^n-\frac{1}{2}y_2^n)^k.$$

Observe that

$$\begin{aligned}\mathbb{P}_i^{(a(c_1c_2R)^n\bar{R})^k}(f) &= \frac{1}{2}x_1^n(\mu_0+\mu_1+\cdots+\mu_{k-1})\\ &= \frac{1}{2}x_1^n\frac{1-(1-\frac{1}{2}x_1^n-\frac{1}{2}y_2^n)^k}{1-(1-\frac{1}{2}x_1^n-\frac{1}{2}y_2^n)}\\ &= \frac{x_1^n}{x_1^n+y_2^n}\cdot(1-(1-\frac{1}{2}x_1^n-\frac{1}{2}y_2^n)^k).\end{aligned}$$

Then there exists some function g such that $\lim_{n\to\infty}(1-\frac{1}{2}x_1^n-\frac{1}{2}y_2^n)^{g(n)}=0$. Also, we have $x_1 > y_2$ if and only if $\lim_{n\to\infty}\frac{x_1^n}{x_1^n+y_2^n}=1$.

After embedding the gadget in Figure 21 in place of the states $\bot$ and $\top$ and replace the letter c_1 with the letters a_1, b_1 from the gadget, and symmetrically c_2 with the letters a_2, b_2, such that the final state of the gadget embedded on the right becomes $\top\top$, the sink state $\top\bot$ and symmetrically the final state of the gadget embedded on the left becomes $\bot\top$ and the sink state $\bot\bot$; Lemma 4.41 concludes the proof of Proposition 4.39 since $1 = val^f(i) > val(i) = 1/2$.

4.8 CONCLUSION

This part of the thesis is characterized by the protagonist player having zero information, this means that he is not aware of the state of the game when making decisions. We dealt with two models, probabilistic finite automata and half-blind games.

The former is a model that has been studied for quite some time now, even though most of the problems are undecidable. When taking the game theory point of view, one of the most important problems is that of value 1: is it true that for every $\epsilon > 0$ there exists some word that is accepted by the automaton with probability at least $1 - \epsilon$. This problem is undecidable [Gimbert and Oualhadj, 2010]. When a problem is undecidable, we are obliged to make some compromise *e.g.* find an interesting subset of instances where the problem is decidable. This was done for a few classes of probabilistic automata. Two classes came up as the most interesting, leaktight automata of [Fijalkow *et al.*, 2012], and simple automata of [Chatterjee and Tracol, 2012]. Curiously, their proofs of decidability used very different techniques.

We demonstrated that leaktight automata are a strict superset of simple automata. This, together with the results from [Fijalkow, 2015], suggests that the notion of leaks captures tightly the complications that cause the undecidability of the value 1 problem. Which serves as motivation to explore whether the notion of leaks can be used in some other, more general, model that is in some sense similar to probabilistic automata.

One possible research direction is to try to use the leak notion to decide some class of partially observable Markov decision processes (one player games with partial information). One result in this direction, for the more restrictive class of #-acyclic POMDPs can be found in [Gimbert and Oualhadj, 2014].

Another research direction is the one that we followed by turning probabilistic finite automata into two player games, by adding a perfectly informed adversary, thereby defining half-blind games. We saw that we can decide the maxmin reachability problem (an analogue of the value 1 problem) for a class of leaktight half-blind games, by using the belief monoid structure, that is a nesting of the Markov monoid one.

Further into this line of research, the decidability of whether $val(s) = 1$ (the minmax reachability problem) and $val^f(s) = 1$ (minmax reachability when Min can use only a finite memory strategy) when the half-blind game is leaktight remains open.

Another problem that we have left open is the exact complexity of the maxmin reachability problem for leaktight half-blind games.

In the case of leaktight probabilistic automata, the value 1 problem is $PSPACE$-complete. The Markov monoid algorithm, that is used to decide it, runs in exponential space. However, the value 1 witness can be guessed, resulting in a

$PSPACE$ upper bound. The matching lower bound comes from a reduction to the problem of intersection of regular languages.

The belief monoid algorithm on the other hand, runs in doubly exponential space, since it is a nesting of the Markov monoid.

BIBLIOGRAPHY

Baier, C. and Grosser, M., 2005. Recognizing omega-regular languages with probabilistic automata. In *20th Annual IEEE Symposium on Logic in Computer Science (LICS' 05)*. doi:10.1109/lics.2005.41.
URL http://dx.doi.org/10.1109/lics.2005.41

Baier, Christel, Bertrand, Nathalie and Größer, Marcus, 2008. *On Decision Problems for Probabilistic Büchi Automata*, pages 287–301. Foundations of Software Science and Computational Structures. Springer Science + Business Media. doi:10.1007/978-3-540-78499-9_21.
URL http://dx.doi.org/10.1007/978-3-540-78499-9_21

Baier, Christel, Größer, Marcus and Bertrand, Nathalie, 2012. Probabilistic omegta-automata. *Journal of the ACM*, 59(1):1–52. doi:10.1145/2108242.2108243.
URL http://dx.doi.org/10.1145/2108242.2108243

Bertoni, A., 1975. *The Solution of Problems Relative to Probabilistic Automata in the Frame of the Formal Languages Theory*, pages 107–112. Lecture Notes in Computer Science. Springer Science + Business Media. doi:10.1007/978-3-662-40087-6_6.
URL http://dx.doi.org/10.1007/978-3-662-40087-6_6

Bertrand, Nathalie, Genest, Blaise and Gimbert, Hugo, 2009. Qualitative determinacy and decidability of stochastic games with signals. In *2009 24th Annual IEEE Symposium on Logic In Computer Science*. doi:10.1109/lics.2009.31.
URL http://dx.doi.org/10.1109/lics.2009.31

Blackwell, David, 1962. Discrete dynamic programming. *Ann. Math. Statist.*, 33(2):719–726. doi:10.1214/aoms/1177704593.
URL http://dx.doi.org/10.1214/aoms/1177704593

Blondel and Canterini, 2003. Undecidable problems for probabilistic automata of fixed dimension. *Theory Comput. Systems*, 36(3):231–245. doi:10.1007/s00224-003-1061-2.
URL http://dx.doi.org/10.1007/s00224-003-1061-2

Brázdil, Tomás, Brozek, Václav and Etessami, Kousha, 2010. One-counter stochastic games. *CoRR*, abs/1009.5636.
URL http://arxiv.org/abs/1009.5636

Bukharaev, R. G., 1980. Probabilistic automata. *Journal of Soviet Mathematics*, 13(3):359–386. doi:10.1007/bf01088986.
URL http://dx.doi.org/10.1007/bf01088986

Chadha, Rohit, Kini, Dileep and Viswanathan, Mahesh, 2014. *Decidable Problems for Unary PFAs*, pages 329–344. Quantitative Evaluation of Systems. Springer Science + Business Media. doi:10.1007/978-3-319-10696-0_26.
URL http://dx.doi.org/10.1007/978-3-319-10696-0_26

Chadha, Rohit, Korthikanti, Vijay Anand, Viswanathan, Mahesh, Agha, Gul and Kwon, Youngmin, 2011. Model checking mdps with a unique compact invariant set of distributions. In *2011 Eighth International Conference on Quantitative Evaluation of SysTems*. doi:10.1109/qest.2011.22.
URL http://dx.doi.org/10.1109/qest.2011.22

Chadha, Rohit, Sistla, A. Prasad and Viswanathan, Mahesh, 2009. *Power of Randomization in Automata on Infinite Strings*, pages 229–243. CONCUR 2009 - Concurrency Theory. Springer Science + Business Media. doi:10.1007/978-3-642-04081-8_16.
URL http://dx.doi.org/10.1007/978-3-642-04081-8_16

Chadha, Rohit, Sistla, A. Prasad and Viswanathan, Mahesh, 2013. *Probabilistic Automata with Isolated Cut-Points*, pages 254–265. Mathematical Foundations of Computer Science 2013. Springer Science + Business Media. doi:10.1007/978-3-642-40313-2_24.
URL http://dx.doi.org/10.1007/978-3-642-40313-2_24

Chatterjee, K., Henzinger, T.A. and Jurdzinski, M., 2005. Mean-payoff parity games. In *20th Annual IEEE Symposium on Logic in Computer Science (LICS' 05)*. doi:10.1109/lics.2005.26.
URL http://dx.doi.org/10.1109/lics.2005.26

Chatterjee, Krishnendu, Doyen, Laurent and Henzinger, Thomas A., 2012. A survey of partial-observation stochastic parity games. *Form Methods Syst Des*, 43(2):268–284. doi:10.1007/s10703-012-0164-2.
URL http://dx.doi.org/10.1007/s10703-012-0164-2

Chatterjee, Krishnendu, Doyen, Laurent, Henzinger, Thomas A and Raskin, Jean-François, 2010. Generalized mean-payoff and energy games. *arXiv preprint arXiv:1007.1669*.

Chatterjee, Krishnendu, Doyen, Laurent, Nain, Sumit and Vardi, Moshe Y., 2014. *The Complexity of Partial-Observation Stochastic Parity Games with Finite-Memory Strategies*, pages 242–257. Lecture Notes in Computer Science. Springer Science + Business Media. doi:10.1007/978-3-642-54830-7_16.
URL http://dx.doi.org/10.1007/978-3-642-54830-7_16

Chatterjee, Krishnendu and Henzinger, Thomas A., 2005. *Semiperfect-Information Games*, pages 1–18. Lecture Notes in Computer Science. Springer Science + Business Media. doi:10.1007/11590156_1.
URL http://dx.doi.org/10.1007/11590156_1

Chatterjee, Krishnendu and Henzinger, Thomas A., 2010. *Probabilistic Automata on Infinite Words: Decidability and Undecidability Results*, pages 1–16. Automated Technology for Verification and Analysis. Springer Science + Business Media. doi:10.1007/978-3-642-15643-4_1.
URL http://dx.doi.org/10.1007/978-3-642-15643-4_1

Chatterjee, Krishnendu, Jurdziński, Marcin and Henzinger, Thomas A., 2003. *Simple Stochastic Parity Games*, pages 100–113. Computer Science Logic. Springer Science + Business Media. doi:10.1007/978-3-540-45220-1_11.
URL http://dx.doi.org/10.1007/978-3-540-45220-1_11

Chatterjee, Krishnendu, Jurdziński, Marcin and Henzinger, Thomas A., 2004. Quantitative stochastic parity games. In *Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '04, pages 121–130. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA. ISBN 0-89871-558-X.
URL http://dl.acm.org/citation.cfm?id=982792.982808

Chatterjee, Krishnendu and Tracol, Mathieu, 2012. Decidable problems for probabilistic automata on infinite words. In *2012 27th Annual IEEE Symposium on Logic in Computer Science*. doi:10.1109/lics.2012.29.
URL http://dx.doi.org/10.1109/lics.2012.29

Clifford, Alfred Hoblitzelle and Preston, Gordon Bambord, 1961. *The algebraic theory of semigroups*, volume 7. American Mathematical Soc.

Colcombet, Thomas, 2009. *The Theory of Stabilisation Monoids and Regular Cost Functions*, pages 139–150. Automata, Languages and Programming. Springer Science + Business Media. doi:10.1007/978-3-642-02930-1_12.
URL http://dx.doi.org/10.1007/978-3-642-02930-1_12

Colcombet, Thomas, 2013. Regular cost functions, part i: Logic and algebra over words. *Logical Methods in Computer Science*, 9(3). doi:10.2168/lmcs-9(3:3)2013.
URL http://dx.doi.org/10.2168/lmcs-9(3:3)2013

Condon, A. and Lipton, R.J., 1989. On the complexity of space bounded interactive proofs. In *30th Annual Symposium on Foundations of Computer Science*. doi:10.1109/sfcs.1989.63519.
URL http://dx.doi.org/10.1109/sfcs.1989.63519

Courcoubetis, C. and Yannakakis, M., 1998. Markov decision processes and regular events. *IEEE Transactions on Automatic Control*, 43(10):1399–1418. doi:10.1109/9.720497.
URL http://dx.doi.org/10.1109/9.720497

Derman, Cyrus, 1962. On sequential decisions and markov chains. *Management Science*, 9(1):16–24. doi:10.1287/mnsc.9.1.16.
URL http://dx.doi.org/10.1287/mnsc.9.1.16

Doyen, Laurent, Massart, Thierry and Shirmohammadi, Mahsa, 2011. *Infinite Synchronizing Words for Probabilistic Automata*, pages 278–289. Mathematical Foundations of Computer Science 2011. Springer Science + Business Media. doi:10.1007/978-3-642-22993-0_27.
URL http://dx.doi.org/10.1007/978-3-642-22993-0_27

Durbin, Richard, Eddy, Sean, Krogh, Anders and Mitchison, Graeme, 1998. Biological sequence analysis cambridge university press. *Cambridge, UK*.

Durrett, Rick, 2010. *Probability: theory and examples*. Cambridge university press.

Fijalkow, Nathanaël, 2015. Profinite techniques for probabilistic automata and the optimality of the markov monoid algorithm. *CoRR*, abs/1501.02997.
URL http://arxiv.org/abs/1501.02997

Fijalkow, Nathanaël, Gimbert, Hugo, Horn, Florian and Oualhadj, Youssouf, 2014. *Two Recursively Inseparable Problems for Probabilistic Automata*, pages 267–278. Mathematical Foundations of Computer Science 2014. Springer Science + Business Media. doi:10.1007/978-3-662-44522-8_23.
URL http://dx.doi.org/10.1007/978-3-662-44522-8_23

Fijalkow, Nathanaël, Gimbert, Hugo, Kelmendi, Edon and Oualhadj, Youssouf, 2015. Deciding the value 1 problem for probabilistic leaktight automata. *Logical Methods in Computer Science*, 11(2). doi:10.2168/LMCS-11(2:12)2015.
URL http://dx.doi.org/10.2168/LMCS-11(2:12)2015

Fijalkow, Nathanael, Gimbert, Hugo and Oualhadj, Youssouf, 2012. Deciding the value 1 problem for probabilistic leaktight automata. In *2012 27th Annual IEEE Symposium on Logic in Computer Science*. doi:10.1109/lics.2012.40.
URL http://dx.doi.org/10.1109/lics.2012.40

Fijalkow, Nathanaël and Kuperberg, Denis, 2014. *ACME: Automata with Counters, Monoids and Equivalence*, pages 163–167. Automated Technology for Verification and Analysis. Springer Science + Business Media. doi:10.1007/978-3-319-11936-6_12.
URL http://dx.doi.org/10.1007/978-3-319-11936-6_12

Fijalkow, Nathanaël, Gimbert, Hugo, Kelmendi, Edon and Kuperberg, Denis, 2016. Stamina (stabilization monoids in automata theory). http://stamina.labri.fr/. Accessed: 2016-07-26.

Gantmacher, 1959. *The theory of matrices vol1*.

Gillette, Dean, 1957. Stochastic games with zero stop probabilities. *Contributions to the Theory of Games*, 3:179–187.

Gimbert, Hugo, 2006. *Jeux Positionnels*. PhD thesis, Université Denis Diderot, Paris.

Gimbert, Hugo, 2007. *Pure Stationary Optimal Strategies in Markov Decision Processes*, pages 200–211. STACS 2007. Springer Science + Business Media. doi:10.1007/978-3-540-70918-3_18.
URL http://dx.doi.org/10.1007/978-3-540-70918-3_18

Gimbert, Hugo and Kelmendi, Edon, 2014a. Note: On A Proof Of Positionality Of Mean-Payoff Stochastic Games. Research report, Université de Bordeaux, LaBRI.
URL https://hal.archives-ouvertes.fr/hal-01091192

Gimbert, Hugo and Kelmendi, Edon, 2014b. Two-player perfect-information shift-invariant submixing stochastic games are half-positional. *CoRR*, abs/1401.6575.
URL http://arxiv.org/abs/1401.6575

Gimbert, Hugo and Oualhadj, Youssouf, 2010. *Probabilistic Automata on Finite Words: Decidable and Undecidable Problems*, pages 527–538. Automata, Languages and Programming. Springer Science + Business Media. doi:10.1007/978-3-642-14162-1_44.
URL http://dx.doi.org/10.1007/978-3-642-14162-1_44

Gimbert, Hugo and Oualhadj, Youssouf, 2014. *Deciding the Value 1 Problem for ♯ -acyclic Partially Observable Markov Decision Processes*, pages 281–292. SOFSEM 2014: Theory and Practice of Computer Science. Springer Science + Business Media. doi:10.1007/978-3-319-04298-5_25.
URL http://dx.doi.org/10.1007/978-3-319-04298-5_25

Gimbert, Hugo, Renault, Jérôme, Sorin, Sylvain, Venel, Xavier and Zielonka, Wiesław, 2016. On values of repeated games with signals. *Ann. Appl. Probab.*, 26(1):402–424. doi:10.1214/14-aap1095.
URL http://dx.doi.org/10.1214/14-aap1095

Gimbert, Hugo and Zielonka, Wieslaw, 2009. Pure and stationary optimal strategies in perfect-information stochastic games.

Gimbert, Hugo and Zielonka, Wiesław, 2004. *When Can You Play Positionally?*, pages 686–697. Lecture Notes in Computer Science. Springer Science + Business Media. doi:10.1007/978-3-540-28629-5_53.
URL http://dx.doi.org/10.1007/978-3-540-28629-5_53

Gimbert, Hugo and Zielonka, Wiesław, 2005. *Games Where You Can Play Optimally Without Any Memory*, pages 428–442. Lecture Notes in Computer Science. Springer Science + Business Media. doi:10.1007/11539452_33.
URL http://dx.doi.org/10.1007/11539452_33

Grädel, Erich, 2004. *Positional Determinacy of Infinite Games*, pages 4–18. Lecture Notes in Computer Science. Springer Science + Business Media. doi:10.1007/978-3-540-24749-4_2.
URL http://dx.doi.org/10.1007/978-3-540-24749-4_2

Gripon, Vincent and Serre, Olivier, 2009. *Qualitative Concurrent Stochastic Games with Imperfect Information*, pages 200–211. Automata, Languages and Programming. Springer Science + Business Media. doi:10.1007/978-3-642-02930-1_17.
URL http://dx.doi.org/10.1007/978-3-642-02930-1_17

Hardy, Godfrey Harold, 2000. *Divergent series*, volume 334. American Mathematical Soc.

Hashiguchi, Kosaburo, 1988. Algorithms for determining relative star height and star height. *Information and Computation*, 78(2):124–169. doi:10.1016/0890-5401(88)90033-8.
URL http://dx.doi.org/10.1016/0890-5401(88)90033-8

Kechris, Alexander S., 1995. *Classical Descriptive Set Theory*. Graduate Texts in Mathematics. Springer New York. doi:10.1007/978-1-4612-4190-4.
URL http://dx.doi.org/10.1007/978-1-4612-4190-4

Kelmendi, Edon and Gimbert, Hugo, 2016. Deciding maxmin reachability in half-blind stochastic games. In *Symposium of Algorithmic Game Theory (to appear)*.

Kirsten, Daniel, 2004. *Distance Desert Automata and the Star Height One Problem*, pages 257–272. Lecture Notes in Computer Science. Springer Science + Business Media. doi:10.1007/978-3-540-24727-2_19.
URL http://dx.doi.org/10.1007/978-3-540-24727-2_19

Kopczyński, Eryk, 2006. *Half-Positional Determinacy of Infinite Games*, pages 336–347. Automata, Languages and Programming. Springer Science + Business Media. doi:10.1007/11787006_29.
URL http://dx.doi.org/10.1007/11787006_29

Kopczyński, Eryk, 2009. *Half-Positional Determinacy of Infinite Games*. PhD thesis, University of Warsaw.

Korthikanti, Vijay Anand, Viswanathan, Mahesh, Agha, Gul and Kwon, YoungMin, 2010. Reasoning about mdps as transformers of probability distributions. In *2010 Seventh International Conference on the Quantitative Evaluation of Systems*. doi:10.1109/qest.2010.35.
URL http://dx.doi.org/10.1109/qest.2010.35

Liggett, Thomas M. and Lippman, Steven A., 1969. Stochastic games with perfect information and time average payoff. *SIAM Rev.*, 11(4):604–607. doi:10.1137/1011093.
URL http://dx.doi.org/10.1137/1011093

Maitra, Ashok P and Sudderth, William, 2012. *Discrete gambling and stochastic games*, volume 32. Springer Science & Business Media.

Martin, Donald A., 1975. Borel determinacy. *The Annals of Mathematics*, 102(2):363. doi:10.2307/1971035.
URL http://dx.doi.org/10.2307/1971035

Martin, Donald A., 1998. The determinacy of blackwell games. *The Journal of Symbolic Logic*, 63(04):1565–1581. doi:10.2307/2586667.
URL http://dx.doi.org/10.2307/2586667

Mashiah-Yaakovi, Ayala, 2014. Correlated equilibria in stochastic games with borel measurable payoffs. *Dynamic Games and Applications*, 5(1):120–135. doi:10.1007/s13235-014-0122-2.
URL http://dx.doi.org/10.1007/s13235-014-0122-2

McIver, A. K. and Morgan, C. C., 2002. *Games, Probability, and the Quantitative μ-Calculus qMμ*, pages 292–310. Logic for Programming, Artificial Intelligence, and Reasoning. Springer Science + Business Media. doi:10.1007/3-540-36078-6_20.
URL http://dx.doi.org/10.1007/3-540-36078-6_20

Mohri, Mehryar, 1997. Finite-state transducers in language and speech processing. *Computational linguistics*, 23(2):269–311.

Nain, Sumit and Vardi, Moshe Y., 2013. Solving partial-information stochastic parity games. In *2013 28th Annual ACM/IEEE Symposium on Logic in Computer Science*. doi:10.1109/lics.2013.40.
URL http://dx.doi.org/10.1109/lics.2013.40

Paz, Azaria, 1971. *Introduction to Probabilistic Automata*. Elsevier BV.

Piccione, Michele and Rubinstein, Ariel, 1997. On the interpretation of decision problems with imperfect recall. *Games and Economic Behavior*, 20(1):3–24. doi:10.1006/game.1997.0536.
URL http://dx.doi.org/10.1006/game.1997.0536

Pin, Jean Eric and Miller, Raymond E, 1986. *Varieties of formal languages*. Plenum Publishing Co.

Puterman, Martin L, 1995. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

Rabin, Michael O., 1963. Probabilistic automata. *Information and Control*, 6(3):230–245. doi:10.1016/s0019-9958(63)90290-0.
URL http://dx.doi.org/10.1016/s0019-9958(63)90290-0

Raskin, Jean-Francois, Chatterjee, Krishnendu, Doyen, Laurent and Henzinger, Thomas, 2007. Algorithms for omega-regular games with imperfect information. *Logical Methods in Computer Science*, 3(3). doi:10.2168/lmcs-3(3:4)2007.
URL http://dx.doi.org/10.2168/lmcs-3(3:4)2007

Schützenberger, M.P., 1961. On the definition of a family of automata. *Information and Control*, 4(2-3):245–270. doi:10.1016/s0019-9958(61)80020-x.
URL http://dx.doi.org/10.1016/s0019-9958(61)80020-x

Shapley, L. S., 1953. Stochastic games. *Proceedings of the National Academy of Sciences*, 39(10):1095–1100. doi:10.1073/pnas.39.10.1095.
URL http://dx.doi.org/10.1073/pnas.39.10.1095

Shargel, Leon, Wu-Pong, Susanna and Yu, Andrew BC, 2007. *Applied biopharmaceutics & pharmacokinetics*. McGraw-Hill.

Simon, Imre, 1990. Factorization forests of finite height. *Theoretical Computer Science*, 72(1):65–94. doi:10.1016/0304-3975(90)90047-l.
URL http://dx.doi.org/10.1016/0304-3975(90)90047-l

Simon, Imre, 1994. On semigroups of matrices over the tropical semiring. *ITA*, 28(3-4):277–294.

Sonin, Isaac M., 2008. *The Decomposition-Separation Theorem for Finite Nonhomogeneous Markov Chains and Related Problems*, pages 1–15. Institute of Mathematical Statistics Collections. Institute of Mathematical Statistics. doi:10.1214/074921708000000264.
URL http://dx.doi.org/10.1214/074921708000000264

Tzeng, Wen-Guey, 1992. A polynomial-time algorithm for the equivalence of probabilistic automata. *SIAM J. Comput.*, 21(2):216–227. doi:10.1137/0221017.
URL http://dx.doi.org/10.1137/0221017

Vardi, Moshe Y. and Wolper, Pierre, 1986. Automata-theoretic techniques for modal logics of programs. *Journal of Computer and System Sciences*, 32(2):183–221. doi:10.1016/0022-0000(86)90026-7.
URL http://dx.doi.org/10.1016/0022-0000(86)90026-7

Williams, David, 1991. *Probability with martingales*. Cambridge university press.

Zielonka, Wieslaw, 2010. Playing in stochastic environment: from multi-armed bandits to two-player games. In *LIPIcs-Leibniz International Proceedings in Informatics*, volume 8. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

Zielonka, Wiesław, 2004. *Perfect-Information Stochastic Parity Games*, pages 499–513. Lecture Notes in Computer Science. Springer Science + Business Media. doi:10.1007/978-3-540-24727-2_35.
URL http://dx.doi.org/10.1007/978-3-540-24727-2_35