INSTITUT
DE RECHERCHE
EN INFORMATIQUE
FONDAMENTALE

Université
de Paris

École doctorale 386 – Sciences Mathématiques de Paris Centre

Monotonic graphs for Parity and Mean-Payoff games

Thèse de doctorat en informatique

par

Pierre Ohlmann

Soutenue publiquement le 13 décembre 2021 devant le jury constitué de

Patricia Bouyer-Decitre	Présidente
Mikołaj Bojańczyk	Rapporteur
Uri Zwick	Rapporteur
Thomas Colcombet	Examinateur
Karoliina Lehtinen	Examinatrice
Sven Schewe	Examinateur
Nathanaël Fijalkow	Directeur de thèse
Olivier Serre	Directeur de thèse

Abstract

In a parity game, Eve and Adam take turns in moving a token along the edges of a directed graph, which are labelled by integers called priorities. This interaction results in an infinite path, and Eve wins the game if the maximal priority appearing infinitely often is even. In the more general setting of mean-payoff games, priorities are replaced by positive or negative integers interpreted as payoffs from Eve to Adam; Eve seeks to minimize their long-term average. Both parity and mean-payoff games are positional: optimal decisions can be made depending only on the current position.

The problems of determining the winner for these two games thus belong to NP $\cap$ coNP, and have attracted considerable attention since the early nineties when parity games were shown equivalent to the model-checking problem for μ-calculus. Both games moreover find numerous practical application, most notably they provide adequate models for synthesis problems on reactive systems.

Despite decades of efforts toward polynomial time algorithms, it was only recently that a breakthrough was achieved in this direction by Calude, Jain, Khoussainov, Li and Stephan, who presented in early 2017 an algorithm running in quasipolynomial time for solving parity games. Quickly after, several different algorithms with similar runtime were discovered, and later unified by the separating approach proposed by Bojańczyk and Czerwiński, and identified as value iteration algorithms.

We introduce monotonic graphs for studying structural and algorithmic aspects of such infinite duration games. These natural objects have numerous (more or less) implicit occurrences in the literature.

We start by showing that the existence of universal well-ordered such graphs characterises (half) positionality of arbitrary winning conditions. This yields a novel approach to establishing and combining such structural results.

We then advocate that (universal) monotonic graphs provide different handles for constructing algorithms. Finite monotonic graphs induce value iteration algorithms, which are shown to be roughly equivalent to Bojańczyk and Czerwiński's separating approach in general. This allows us to formulate lower bounds for mean-payoff games, and conclude that value iteration algorithms are inadequate to improve on the current state of the art. We also study value iteration algorithms for different well-known extensions of these games.

Monotonic graphs also give a generic formalisation for strategy improvement algorithms. More precisely, we establish that valuations induced by monotonic graphs are fit for strategy improvement if and only if they are positional for the opponent. This encompasses known strategy improvement frameworks, allows us to propose new algorithms and perhaps more importantly, introduces a new tool for their difficult study.

Surprisingly, monotonic graphs also find applications for symmetric algorithms, such as those based on attractors. For parity as well as mean-payoff games, we find that monotonic graphs allow us to shed light and improve on the recent state of the art.

Résumé

Dans un jeu de parité, Eve et Adam déplacent tour à tour un jeton le long d'un graphe dirigé dont les arêtes sont étiquetées par des entiers appelés priorités. Cette interaction produit un chemin infini ; Eve remporte la partie si la plus grande priorité apparaissant infiniment souvent est paire. Dans le cadre plus général offert par les jeux à paiement moyen, les priorités sont remplacées par des entiers potentiellement négatifs représentant des paiements d'Eve à Adam. Eve cherche donc à minimiser leur moyenne à long terme. Les deux types de jeux sont positionnels : des décisions optimales peuvent être prises en fonction seulement de la position actuelle.

Le problème de déterminer le gagnant dans ces deux jeux se situe donc à l'intersection de NP et de coNP. Ces questions algorithmiques sont l'objet d'une attention considérable depuis le début des années 1990, au moment où il a été établi que les jeux de parité sont équivalents au problème de la vérification pour la logique du mu-calcul. Les deux jeux ont de nombreuses applications pratiques ; ils fournissent notamment des modèles adéquats pour le problème de la synthèse de systèmes réactifs.

Malgré des dizaines d'années de recherche d'algorithmes fonctionnant en temps polynomial, c'est seulement en 2017 que le premier algorithme quasipolynomial pour les jeux de parité a été découvert par Calude, Jain, Khoussainov, Li et Stephan. Peu après, plusieurs autres algorithmes quasipolynomiaux pour les jeux de parité ont été présentés, puis unifiés grâce à l'approche de séparation proposée par Bojanczyk et Czerwinski, et enfin identifiés comme des algorithmes d'itération de valeur.

Nous introduisons les graphes monotones dans le but d'étudier les aspects structurels et algorithmiques des jeux à durée infinie. Ces objets naturels ont fait de nombreuses apparitions (plus ou moins) implicites dans la littérature.

Nous montrons en premier lieu que, pour des conditions de gain arbitraires, l'existence de graphes monotones universels bien ordonnés caractérisent la positionnalité pour Eve. Cela donne une nouvelle technique pour établir et combiner de tels résultats structurels.

Nous avançons ensuite que les graphes monotones offrent différentes possibilités pour construire des algorithmes. Les graphes monotones finis induisent des algorithmes d'itération de valeur, dont on montre qu'ils sont équivalents dans un cadre général à l'approche (forte) de séparation de Bojanczyk et Czerwinski. Cela nous permet en particulier de formuler des bornes inférieures pour les jeux à paiement moyen, et donc d'établir que les méthodes d'itération de valeur ne peuvent améliorer l'état de l'art. Nous étudions aussi les algorithmes d'itération de valeur pour différentes extensions courantes de ces jeux.

Les graphes monotones donnent aussi un cadre générique pour formuler des algorithmes d'amélioration de stratégies. Plus précisément, nous montrons que les valuations induites par des graphes monotones permettent de tels algorithmes si et seulement si elles sont positionnelles pour l'adversaire. Ce résultat capture les différents cadres connus, nous permet d'en proposer d'autres, et introduit un nouvel outil à l'étude difficile de ces algorithmes.

Étonnament, les graphes monotones s'appliquent aussi à l'étude d'algorithmes symétriques, tels que ceux qui sont fondés sur des calculs d'attracteurs. Ils permettent d'envisager sous un nouvel angle les jeux de parité ainsi que les jeux à paiement moyen, et dans les deux cas, de mieux comprendre et d'améliorer l'état de l'art.

Acknowledgments

First, I would like to thank my two PhD advisers Nathanaël Fijalkow and Olivier Serre. Nath, I met you in my first research internship in 2016 when I had barely heard about Turing machines. We have never really stopped interacting ever since. I am deeply grateful to you for introducing me to so many fascinating problems (roughly, each one I know) and to so many researchers (roughly, everyone I know and among which Olivier). Olivier, I am so grateful to you for always being so present, kind and supportive. Experience has shown on many occasions that one should always follow an advice from Olivier! I have learned so much from the two of you, and I cannot imagine what these years would have been without you by my side. Special thanks also go to Thomas Colcombet for being such a friendly, passionate and inspiring PhD grandfather.

Second, I want to thank Mikolaj Bojańczyk and Uri Zwick for accepting to evaluate my work, and extend this gratitude to Patricia Bouyer, Karoliina Lehtinen and Sven Schewe. It is an immense pleasure for me to have the eight of you as my jury. I have found a lot of pleasure and inspiration in reading you; I hope that you in turn will enjoy reading my thesis, and that this will sprout further discussions to which I am very much looking forward.

Third I would like to thank all the other people I have had the pleasure to collaborate with, Antonio, Guillaume, Thejaswini, Sasha, Marcin, Rémi, Denis, Michał, Mahsa, Arnaud, Paweł, Joël, Amaury, Engel and others. I deeply regret that working together was made so much more difficult by the pandemic. I also thank my old mates Simon and Vincent (from afar), my new mate Olisti, and all the fellow students I have met at conferences and events.

Last, I want to thank my closest friends and my family for their presence and support. Thank you Pricou, Loulou, Mazzu, Maud, Mad, Lélé, Coll, Chan, PLFzoinzoin and the others. Special thanks go to my grandparents for introducing me to computers as a kid, and of course to my parents and siblings for always being there. Above all, I am grateful to Lara. In so many ways this thesis is the fruit of your love and kindness over the years.

Table of contents

General introduction

Overview

We introduce parity and mean-payoff games, discuss their significance and then give a high-level description of our approach. Although the discussion is non-technical, it is nonetheless quite dense, with a focus on describing related works and motivations as thoroughly as possible.

We refer the reader not familiar with infinite duration games to the preliminaries below, which provide a gentle introduction to all the concepts needed throughout the thesis. Apart from the somewhat demanding opening part, we have paid a special attention so as to make the overall exposition self-contained and accessible.

1 Parity games

In a parity game, two players, Eve and Adam, push a token along the edges of a directed graph (without dead-ends). This interaction goes on forever, producing an infinite path π. The edges of the graph are coloured with integers called *priorities* and used by the *parity winning condition* defined as follows: π is winning for Eve if and only if the largest priority appearing infinitely often in π is even. A fundamental property of parity games, which we will discuss at length, is their bi-positional determinacy: if either player can ensure to win then they can do so with a strategy which depends only on the current vertex of the graph.

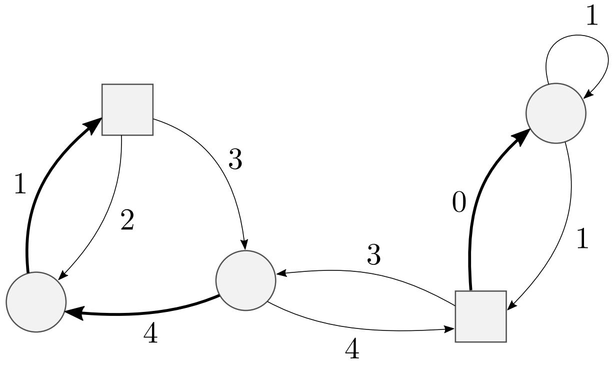


Figure 1: Example of a parity game. Circle vertices are controlled by Eve, and square vertices are controlled by Adam (this convention is used throughout the thesis). The bold edges represent winning positional strategies: from the three leftmost vertices, Eve can ensure a win, whereas Adam wins from the two vertices on the right.

A few other related winning conditions are discussed below and will be formally defined in the preliminaries for completeness.

1.1 Origins of parity games

Parity games originate from automata theory and μ-calculus; we now give a brief history describing the context of their apparition and some of the landmark publications. We refer to [GTW02] for a complete presentation of all results mentioned below.

Rabin's complementation lemma. The theorem of Rabin [Rab69] states that the monadic second order logic over infinite binary trees (S2S) is decidable. This result is of utmost importance in modern computer science and logics, and is often referred to as the "mother of all decidability results". Rabin's theorem is proved by studying infinite tree automata and most crucially, showing that these admit effective complementation. The proof of Rabin is notoriously difficult, and simplifying it has been an important challenge during several decades.

The fundamental idea of using infinite duration games in this context was first suggested by Büchi [Büc77], then succesfully implemented by Gurevich and Harrington [GH82] and independently by Muchnik [Muc84]. This approach relies on proving finite-memory determinacy (winning strategies can be implemented by finite-state machines) of infinite games[1] with a Muller winning condition, which are more general than parity games.

Simplifications of the approach of [GH82] were given by Yakhnis and Yakhnis [YY90] and then by Zeitman [Zei94], who in particular considered the case of infinite games which are not necessarily played over trees. Although playing over infinite graphs or trees is roughly equivalent in this case, this suggests playing (with infinite duration) over finite graphs, which was first investigated by McNaughton in his seminal paper [McN93] for the Muller condition.

The μ-calculus. The μ-calculus extends propositional modal logics by adding fixpoint operators. It originated in the work of Scott and de Bakker [SB69] and was subsequently developped by many different authors; the μ-calculus as we know it today was formalised by Kozen in his seminal paper [Koz83].

The μ-calculus is used to describe and verify properties of labelled transition systems. It is known to be very expressive and encodes most modal logics such as Hennessy and Milner's dynamic logic HML [HM80] and several temporal logics (for instance CTL^*, further discussed below). At the same time it enjoys good algorithmic properties, making it a central logic in modern verification. Its prolific mathematical theory is rooted within (finite) model theory. A relationship between the μ-calculus and infinite tree automata (all automata discussed below operate over infinite trees) was first laid out in [SE84], where, in order to establish its membership in EXPTIME, the satisfiability problem is reduced to the emptiness problem for a class of automata.

In [Niw86] and [Niw88], Niwińkski further investigated this correspondence. The articles present a reduction from automata to μ-calculus, and a converse reduction in the absence of conjuncts. In their celebrated publication [EJ91], Emerson and Jutla describe a complete converse reduction establishing an (effective) equivalence in expressivity between the two formalisms, thus giving yet an alternative proof of the complementation lemma (since the μ-calculus admits easy complementation). The new proof consists in first translating a μ-calculus formula to an alternating automaton (as introduced by Muller and Shupp in [MS87]) with a Streett winning condition, and then applying the co-Safra construction from [EJ89] (based on [Saf88]) which yields a non-deterministic Rabin automaton.

Most importantly, Emerson and Jutla show that combining the two above translations yields a Rabin automaton whose winning condition is much simpler and in fact coincides with the parity

[1] Finite-memory determinacy of finite such games was established by Büchi and Landweber [BL69] and also constitutes a milestone of early automata theory, discussed below.

condition. They give a first study of parity games (over infinite trees), and present a direct determinacy proof based on its μ-calculus formulation. They also formulate a concise and elegant proof of positionality which considerably simplifies the proofs of finite-memory determinacy of Muller games discussed above. Prior to their work, parity automata were also studied by Mostowski in [Mos84] and shown to be equivalent in expressivity to Rabin or Muller automata without appealing to games or to the μ-calculus. Positionality of infinite parity games was also proved in [Mos91], independently of [EJ91].

1.2 Significance and motivations

Model checking μ-calculus. The model checking problem asks, given a specification (here, a μ-calculus formula) and a model (a finite labelled transition system), whether the formula holds over the model. As explained above, the works of Niwiński and Emerson and Jutla together established equivalence in expressiveness between tree automata with parity acceptance on one hand and the μ-calculus on the other. This was made more precise by Emerson, Jutla and Sipsa in [EJS93] (see also the full version [EJS01]) who provided a linear equivalence between the model checking problem for the μ-calculus and the emptiness problem for automata with parity conditions, which is easily seen to be equivalent to solving finite parity games. Positionality of finite parity games (which can already be established as a consequence of [Eme85]) then gives membership of the problem in NP $\cap$ coNP. By reduction to discounted games, Jurdziński [Jur98] established that the problem also belongs to UP $\cap$ coUP.

The equivalence with μ-calculus model checking as well as their intriguing complexity status give two excellent theoretical motivations for studying the complexity of solving finite parity games. The attention of the practical model checking community has however considerably diverged from the μ-calculus; most modern practical applications of parity games are related to synthesis rather than verification.

Synthesis of reactive systems. Reactive systems are those which maintain an ongoing interaction with their environments. Examples include embedded controllers, hardware circuits, communication protocols, distributed systems, and many more. Church was the first to pose in [Chu57] the question of synthesis: given an input/output specification, is it possible to synthesise a program that meets the specification? Synthesis of reactive systems is a field of its own, which has recently seen tremendous developments; we give a few early landmarks and refer to the surveys of Finkbeiner [Fin16] and of Bloem, Chatterjee and Jobstmann [BCJ18] for more complete and exhaustive expositions.

Infinite duration games have quickly emerged (earliest appearances date back to the work of McNaughton, see [McN67]) as the natural model underlying synthesis for reactive systems. In this scenario, the two players model respectively the system, which tries to satisfy the specification, and an adversarial environment, aimed at breaking the specification. A solution for the synthesis problem then corresponds to a strategy for the system player which can be implemented by a finite state machine (or program).

Decidability of the synthesis problem was established by Büchi and Landweber [BL69], while the infinite tree automata of Rabin [Rab69; Rab72] provided an algorithmic formalism for synthesised strategies (or programs). In this early framework, system specifications are given in monadic second order logic which has unpractical non-elementary complexity.

Practicality of the synthesis problem became conceivable with the development of (weaker) temporal specification logics initiated by Pnueli's linear temporal logic [Pnu77] (LTL). Emergence of LTL was quickly followed by that of the branching time computation tree logic (CTL) of Ben-

Ari, Manna and Pnueli [BAMP81] and (independently and roughly equivalently) of Clarke and Emerson [CE81]. These two logics are standardly used in the synthesis and model checking community, and subsumed by the logic CTL* of Emerson and Halpern [EH83] and by the μ-calculus of Kozen [Koz83].

LTL model checking can be done in PSPACE (the same applies to CTL*) and became an important industrial technique already in the eighties in the context of hardware design. The problem of LTL synthesis was shown to be 2EXPTIME-complete in the seminal work of Pnueli and Rosner [PR89], who also defined the automata-theoretic approach (originated from Buchi and Landweber's work [BL69]) to the synthesis problem. The difference in complexity between verification and synthesis should be tempered by the fact that the (generally exponential) model is part of the input in the model checking problem.

The fast-paced development and success of the field of automatic verification at that time (which continues today) motivated a lot of research on reactive synthesis and its automata theoretic foundations: see for instance the works of Nerode, Yakhnis and Yakhnis [NYY92], of Thomas [Tho95] and of Vardi [Var95]. Despite thorough efforts, important theoretical developments and a more and more mature underlying theory, synthesis of reactive systems started becoming a reality only in more recent times.

Modern days. Fragments of LTL captured by parity games with a small fixed number of priorities (which can be solved in polynomial time) have been studied, most prominently the generalised reactivity GR(1) of Piterman, Pnueli and Sa'ar [PPS06], which generalises most fragments studied earlier. Formulas in GR(1) translate to games (of exponential size) which admit only three priorities. For the first time, small industrial reactive designs (from [Spe99]) could be synthesised (see [BJP+12]).

Another early successful approach to LTL synthesis is the bounded synthesis framework of Schewe and Finkbeiner [SF07], which obtains tractability for many instances of the full LTL synthesis (and can be applied to other formalisms). Roughly speaking their technique builds on the safraless determinisation of Kupferman and Vardi [KV05] and explores the space of programs by iteratively incrementing a bound on their maximal size. The framework of bounded reactive synthesis has become standard, and is often used in combination with symbolic methods such as binary decision diagrams (which proved successful in model checking) for dealing with large state-spaces, and/or with SAT or SMT-solvers.

Perhaps surprisingly, the recent years have witnessed a resurgence of LTL reactive synthesis tools based on novel automata theoretic translations combined with explicit parity game solving, as opposed to combinations of bounded synthesis with symbolic approaches. Most notably, the tool STRIX of Meyer, Sickert and Luttenberger (see [MSL18] and [LMS20]), which has won the main synthesis competition SYNTCOMP each year since 2018, is based on such methods. First, the LTL formula is converted (using the library Owl which implements many recent efficient automata-theoretic translations, see for instance the work of Esparza, Křetínský and Sickert [EKS18]) into a parity game[2]. The game is then solved using a strategy improvement algorithm of Luttenberger [Lut08] implemented over GPUs.

Kupferman explains in [Kup12] that the reasons for lack of practical impact (at that time) of reactive synthesis are not only algorithmic (non-trivial algorithms implemented on implicit or explicit parity games resulting from intricate determinisation procedures), but also methodological: standard automata theoretic approaches often lack in modularity and flexibility. Many efforts in the

[2] The reality is slightly more intricate: several deterministic parity games are obtained from different subformulas in a well-chosen decomposition, which are then composed into a larger parity game.

synthesis community have been devoted to addressing such issues, and tools and frameworks have been developed in the recent years which are more and more robust, applicable, and scalable.

1.3 Three families of algorithms

For the reasons detailed above, and also because problems belonging to NP ∩ coNP have (often after considerable efforts) generally been proved to be solvable in polynomial time, finite parity games have attracted a lot of attention since the mid-nineties; they are however still not known to be solvable in P. The only breakthrough in this direction was obtained by Calude, Jain, Khoussainov, Li and Stephan [CJK+17] who presented an algorithm with quasipolynomial runtime $O(n^{\log d})$, where we use n and d to respectively denote the size and number of different priorities appearing on the game. It should be noted that in all practical applications (except model checking μ-calculus), the size is typically exponential in the number of priorities; in this case the algorithm of [CJK+17] runs in polynomial time.

The discussion below is far from being exhaustive and many important contributions relative to solving parity games will not be mentioned. We focus on three well-established and important classes of algorithms, namely *value iteration*, *attractor-based* and *strategy-improvement* algorithms. These three paradigms are central to our work, and will be discussed in more depth respectively in Chapters 4, 9 and 11.

Value iterations. The first value iteration algorithm for parity games is due to Jurdziński [Jur00]. It is based on successive updates of $d/2$-tuples of integers, one for each vertex, representing occurrences of odd priorities which can be forced by Adam, and ordered lexicographically. This technique is rooted in Walukiewicz's *signatures* [Wal96] which are implicit in the work of Emerson and Jutla [EJ91] and instrumental in the study of infinite parity games.

Its worst-case complexity is roughly $n^{d/2}$, which was already obtained by earlier (arguably more complicated) μ-calculus model checking algorithms. However its polynomial space complexity was only matched at that time by Zielonka's attractor-based algorithm (discussed below), making it the most efficient algorithm in theory at the time of its introduction, as well as one of the most conceptually simple. In practice however, it is well-known to behave badly, and even with known optimisations exponential runtime is frequently displayed.

Schewe [Sch07] was the first to restrict the domain of the tuples in his algorithm inspired by the big-step attractor-based approach of [JPZ06], further discussed below. This brought down the worst-case complexity to roughly $n^{d/3}$, which was (in theory) the best algorithm available until 2017 in the typical case where $d = o(\sqrt{n})$.

Within a few months following the breakthrough of [CJK+17], two different quasipolynomial value iteration algorithms were given by Fearnley, Jain, Schewe, Stephan and Wojtczak [FJS+17] (see also [FJK+19]) and by Jurdziński and Lazić [JL17]. Both algorithms reduce the space complexity to quasilinear, and both papers provide additional analyses of their (very similar) runtime bounds. The first one is closer to the approach of [CJK+17] and uses a similar data structure, whereas the second one is based on an elegant tree-coding lemma and directly refines [Jur00], drastically reducing the domain of the valuation to a quasipolynomial size of $n\binom{\log n+d/2}{d/2}$.

A year later, Lehtinen [Leh18] (see also [LB20]) presented a fourth quasipolynomial algorithm, based on a novel notion of register-index of a parity game. Register-indices were then used by Boker and Lehtinen [BL18], who generalised the approach to the setting of alternating parity word automata, showing that they can be turned into alternating weak automata with only quasipolynomial blow-up[3]. Although more general, the algorithm of Lehtinen displayed a slightly worse quasipoly-

[3] When instantiated with only one letter, this yields a quasipolynomial reduction from parity to safety games, which

nomial runtime of roughly $n^{\log d \log n}$; the above translation was later improved (using universal trees, discussed below) by Daviaud, Jurdziński and Lehtninen [DJL19] so as to match the quasipolynomial complexity of the other algorithms when instantiated to parity games.

Meanwhile, Bojańczyk and Czerwiński [BC18] formalised the data structure of [CJK+17] as a deterministic (strongly) separating automaton of quasipolynomial size, and explained that any deterministic separating automaton implies a reduction to a safety game of roughly the same size, and therefore an efficient algorithm. Independently, Fijalkow [Fij18] presented the tree-coding lemma of [JL17] as a construction of a universal tree, showed that any universal tree gives rise to a value iteration algorithm, and established an almost matching (up to a polynomial factor) lower bound on the size of universal trees.

In a combined effort, Czerwiński, Daviaud, Fijalkow, Jurdziński, Lazić and Parys [CDF+18] unified all above results by showing that any (even non-deterministic) strongly separating automaton includes a universal tree, and that all algorithms above implicitly or explicitly construct universal trees. This second result was already established by [BC18] for the data structure of [CJK+17] and thus for [FJS+17] which is closely related, and is relatively straightforward for universal trees. However simulating Lehtinen's algorithm register games with a separating automata turned out to be quite technical, and produces an automaton which is not deterministic and therefore not fit for a game reduction. The results of [CDF+18] established nonetheless a quasipolynomial *combinatorial barrier* in the form of universal trees which applies to all quasipolynomial approaches known at that time. Later, Parys [Par20] has completed the picture with its missing piece, by showing that the separating automata implicit in Lehtinen's algorithm can indeed be applied in a game reduction scenario, as was claimed in [CDF+18].

Attractor-based algorithms. Attractor-based algorithms originated in McNaughton's simplification [McN93] of Gurevich and Harrington's approach [GH82] when applied to the case of *finite* Muller games. Zielonka[4] [Zie98] was the first to instantiate it to parity games, which led to the so-called *Zielonka algorithm*. It has a recursive nature, and is based on the following simple steps illustrated in Figure 2. (We recall that d is the maximal priority, which is assumed to be even).

1. Determine in linear time the set A of vertices from which Eve can ensure to see the priority d (note here that by definition, Adam can ensure from ${}^{c}A$ to remain in ${}^{c}A$), A is called the *Eve-attractor* to priority d.

2. Recursively solve the game G' obtained by removing A and edges of priority d.

3. If Adam's winning region W'_{Adam} in G' is empty, then G is everywhere winning for Eve. Otherwise by the above remark the Adam-attractor B to W'_{Adam} is (non-empty and) winning for Adam in the original game G and can therefore safely be removed.

It is easy to see that the algorithm runs in time at most n^d and examples of parity games displaying an exponential runtime are known since at least [Jur00]. More recently in the work of Gazda and Willemse [GW13], single-player examples were given over which the algorithm also exhibits an exponential runtime. Despite having exponential worst-case runtime, Zielonka's algorithm is well known to perform very well in practice. In fact, it has been observed by Friedmann

can be solved in linear time. Boker and Lehtinen also considered the case of alternating tree automata for which they provided a lower bound.

[4] Zielonka's purpose was to present an alternative to Emerson and Jutla's positionality proof which is given over infinite trees [EJ91]. His proof applies to any infinite game graph, turning his induction into an algorithm for finite graphs is transparent. Although simpler (it is, in our opinion, the simplest proof), a drawback in Zielonka's approach is that both players are involved in the inductive argument.

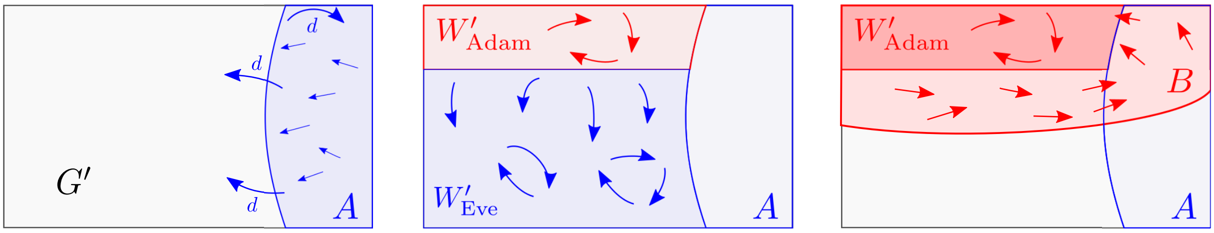


Figure 2: Illustration of the three steps, from left to right, in the case where $W'_{\text{Adam}} \neq \varnothing$ (otherwise, the algorithm terminates). The blue and red arrows respectively represent positional strategies for Eve and Adam. In the second step, the strategies are winning in G' by induction, and in the third step, Adam's strategy is winning is G; Eve's strategy is discarded in this case.

and Lange [FL09] and more recently by van Dijk [Dij18b] that even with very few optimisations, Zielonka's algorithms consistently outperforms all others on both randomly generated games and benchmarks from practical applications.

The strength and appeal of attractor-based algorithms lie in their simplicity and modularity. An important example is in the approach of Jurdziński, Paterson and Zwick [JPZ06] who modified Zielonka's algorithm by adding a brute-force search for small dominia[5]. The obtained algorithm has runtime roughly $n^{\sqrt{n}}$, making it the first deterministic subexponential time algorithm, and the only one available until 2017. The work of Gajarský, Lampis, Makino, Mitsou and Ordyniak [GLM+15] presents many different adaptations of the original attractor-based algorithm to run in polynomial time with respect to several parametrizations[6]. Other notable examples of attractor-based algorithms include the priority promotion of Benerecetti, Dell'Erba, and Mogavero [BDM16] and the tangle-learning scheme of van Dijk [Dij18a].

Recently, and most relevant to our work, Parys [Par19] introduced yet another quasipolynomial time algorithm in the form of a surprisingly simple modification of Zielonka's algorithm. Roughly, Parys proves that it suffices to guide the recursive calls with two additional integer parameters (one for each player), which in particular force the runtime to be only quasipolynomial. The crux of the proof still lies in a separation result: partitions returned by the modified recursive calls no longer necessarily correspond to the winning regions of the two players, but they separate dominia of adequate sizes, which is sufficient to obtain correctness. Despite this fact, Parys' approach seemingly avoids the combinatorial barrier imposed by separating automata, because its main mechanic directly operates on the structure of the graph ; it is not clear how to describe the algorithm as implicitly constructing an automaton.

However the community quickly realised that universal trees still seemed hidden in the new approach. Lehtinen, Schewe and Wojtczak [LSW19] presented another attractor based approach, which directly combines the ideas of Parys with the (essentially optimal) universal tree of [JL17]. Besides lowering the complexity to roughly the square of the value iteration algorithm, this strongly suggests that universal trees are inherent also to the new approach, subjecting it to the same lower bound. Jurdzinski and Morvan [JM20] then proved that the attractor-based approach can be instantiated with any universal tree, and moreover provided a symbolic description of the approach.

Unfortunately, these new quasipolynomial algorithms do not share the efficiency of Zielonka's on practical instances as explained by Lehtinen, Parys, Schewe and Wojtczak in their recent joint

[5] Dominia are subgames in which a player can ensure to win from everywhere.

[6] Since Obdrzálek [Obd03; Obd06; Obd07] showed fixed parameter tractability over graphs of bounded tree-width, and later clique-width and DAG-width, such questions have attracted considerable attention, but we will not discuss them further.

paper [LPS+21].

Strategy improvements. The strategy improvement paradigm has a long history, which is rooted in stochastic processes and games. As the name suggests, starting from an arbitrary (positional) strategy, one iteratively computes "better and better" strategies until reaching an optimal one. This possibility of running a strategy improvement scheme relies on being able to *evaluate* a strategy, in such a way that one may efficiently compute a better strategy from any non-optimal one. In a similar way as for the simplex method in linear programming, specifying a strategy improvement algorithm requires specifying a (potentially randomised) way of choosing an improved strategy; this is usually called an *improving policy* or a *switching policy*.

The strategy improvement framework was introduced in the context of Markov decision processes by Howard [How60] and later generalised to Shapley's [Sha53] stochastic games by Hoffman and Karp [HK66] and Rao, Chandrasekaran and Nair [RCN73]. In this context, the evaluation of the strategy is naturally suggested by the definition: simply use the values in the process induced by fixing the strategy. Important landmarks in the vast literature concerning strategy improvements in stochastic contexts (often called policy iteration) include the work of Ludwig [Lud95], who adapted Bland's rule [Bla97] and extended Kalai's analysis [Kal92] from linear programming to Condon's simple stochastic games [Con92], leading to a randomised improvement policy with subexponential runtime $2^{O(\sqrt{n})}$. Much later, Ye [Ye11] proved a strongly polynomial upper bound for Markov decision processes when the discount factor is fixed, which was improved and generalised to stochastic games by Hansen, Miltersen and Zwick [HMZ13] (for Dantzig's rule or "single-switch").

The observation that the strategy improvement paradigm can be applied to parity games (via reduction to discounted games[7], which are a special case of stochastic games) is due to Puri [Pur95]. It was first made explicit as a completely combinatorial approach (formally removing the need to translate to discounted games) by Vöge in his PhD thesis [Vög00] (in German), and considerably popularised by the seminal paper of Vöge and Jurdziński [VJ00]. In this scenario, evaluations of strategies are much more involved (at least conceptually): one should compute an optimal counter-strategy, and precisely inspect the (ultimately cycling) paths in the induced finite graph.

Despite these technical complications, the combinatorial approach lends itself to implementations, and sparked a lot of excitement in the community. The number of iterations turned out to be consistently sublinear on benchmarks, and strategy improvements were usually regarded (until the surprising observation of [FL09] that Zielonka's algorithm is actually more robust) to be the most practical algorithms. Most importantly, the striking absence of lower bounds[8] was what made the approach to be widely considered as a contender for a polynomial time algorithm.

An important series of papers from Björklund, Sandberg and Vorobyov [BSV03; BSV04a; BV05], Ludwig [Lud95]'s aforementioned work and further discussed below for mean-payoff games, is devoted to applying both randomised pivoting rules of Kalai [Kal92] and of Matoušek, Sharir and Welzl [MSW96] as switching policies to the strategy improvement framework. This provided the first subexponential algorithm, which was randomised, with runtime $2^{O(\sqrt{n \log n})}$.

Later, Schewe [Sch08] presented a novel (combinatorial) framework, in which improvements are locally optimal and yet still computed in polynomial (actually, even slightly superlinear) time by solving an adequate two-player game. This surprising development, supported by Schewe's observation that even fewer iterations are performed, strengthened the belief that strategy improvements could be proved to run in polynomial time. Luttenberger [Lut08] gave an alternative presentation

[7] The reduction from parity to discounted games goes through mean-payoff games; the second step (from mean-payoff to discounted) is due to Zwick and Paterson [ZP95; ZP96].

[8] A lower bound was known [BV05] but is was unsatisfactory since switches are chosen adversarially, and not according to one of the known natural policies.

of Schewe's algorithm as a *non-deterministic strategy* improvement scheme directly adapted from that of Björdklund, Sandberg and Vorobyov, and showed that over the slightly particular parity games considered by the algorithm, the formalism actually coincides with the original one of [VJ00], imported from discounted games. It is also worth mentioning that a variation on Luttenberger's algorithm based on non-deterministic strategies and implemented on the GPU is a key component in STRIX's [LMS20] LTL synthesis tool.

Friedmann's breakthrough result [Fri09] consists in a notoriously involved construction of a parity game displaying an exponential number of iterations for the most natural improvement policy. Friedmann also explains how to slightly modify the game so as to adapt the lower bound to Schewe's improvement scheme. Friedmann's counter examples later proved to be extremely robust and modular: these were adapted (by Friedmann and co-authors) to several different scenarios, including (but not limited to) non-oblivious policies for parity games [Fri11a; Fri13], and to the most common randomised pivoting rules for the simplex algorithm, along with Hansen and Zwick [FHZ11].

Although this direction was still advocated by Friedmann himself [Fri11b], efforts for developing a polynomial time strategy improvement algorithm have considerably declined since then. A notable exception is the work of Schewe, Trivedi and Varghese [STV15] which proposes a symmetric strategy improvement scenario, where strategies for each player are improved in parallel, and influence each other in the chosen switches. The empirical runtime of the symmetric algorithm is encouraging, even on different variants of Friedmann's examples. To the best of our knowledge, no lower bounds are known.

To date, no quasipolynomial strategy improvement algorithm is known. Crafting such an algorithm appears to be an interesting but challenging endeavour for at least two reasons. First, all lower bounds induced by Friedmann's constructions are exponential or at least subexponential, because based on incrementing a binary counter, and therefore a quasipolynomial algorithm would inherently transcend this barrier. Second, all known strategy improvement algorithms (including the more recent symmetric algorithm of [STV15] or the snare-based non-oblivious scheme of Fearnley [Fea10a]) are applicable in the more general setting of mean-payoff games, therefore such a quasipolynomial strategy improvement would either be specific to parity games (which would be extremely interesting in itself), or imply a new breakthrough for mean-payoff games.

2 Mean-payoff games

We now give a similar treatment to mean-payoff games: we quickly introduce the formalism (and the closely related energy games), then describe their origins and state of the art, their significance, and common extensions.

Mean-payoff and energy games. In a mean-payoff game, edges are labelled by integer weights in $[-N, N]$, which are interpreted as payoffs from Eve to Adam; negative payoffs then correspond to gains for Eve. Eve seeks to minimize the long term average payoff $\limsup \frac{1}{k}\sum_{i=0}^{k-1} t_i$ of the infinite sequence $t_0, t_1, \ldots$ of weights which are seen along the visited path.

Closely related are energy games, which are played over the same kind of graphs, but inherently refer to the evolution of a quantity which should remain non-negative, such as an amount of energy. Starting from a given initial energy n, Adam[9] should ensure that the accumulated energy remains above zero (the battery is never depleted), formally all partial sums $n + \sum_{i=0}^{k-1} t_i$ should be $\geqslant 0$.

[9] We will later prefer to take Eve's point of view and adopt another convention; the definition given here is better aligned with the literature.

Mean-payoff and energy games are *determined*: an optimal *value* can be associated to each vertex in the graph. For mean-payoff games, the value of a vertex belongs to $[-N, N]$ and corresponds to the average payoff from Eve to Adam assuming both players play optimally; in an energy game, the value belongs to $[0, \infty]$ and corresponds to the minimal n (which is ∞ if there is none) such that Eve can ensure to win the game with initial energy n. Just like parity games, finite mean-payoff and energy games are positionally determined for both players. A complete example is discussed in Figure 3.

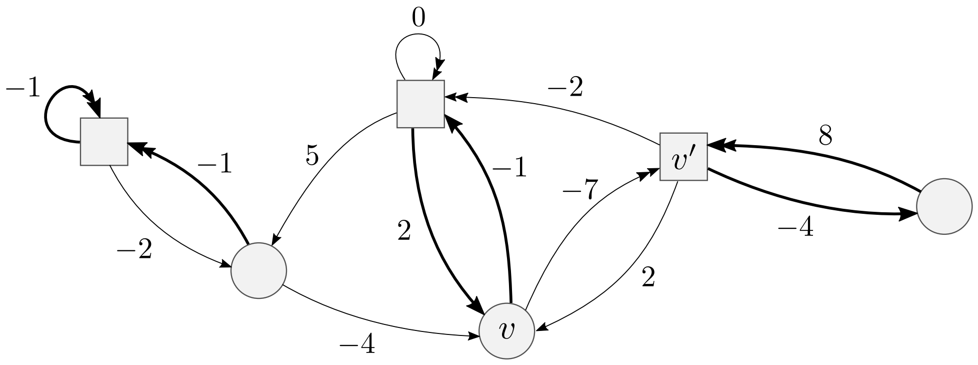


Figure 3: Example of a mean-payoff game. Mean-payoff values from left to right are $-1, -1, \frac{1}{2}, \frac{1}{2}, 2$ and 2, and mean-payoff-optimal positional strategies for both players are identified in bold. Energy values are $\infty, \infty, 0, 9, 2$ and 0, and energy-optimal strategies are given by arrows with double heads. Notice that starting with energy 8 from v, Eve can ensure to deplete the battery.
However to do so she must take the edge towards v', which is non-optimal with respect to mean-payoffs: it gives Adam the possibility to ensure a long term average of 2 by forcing the rightmost cycle.
Notice also that using a mean-payoff-optimal strategy from v' ensures a win for Adam in the energy game with initial energy 4. Actually, mean-payoff-optimal strategies for Adam are also viable in the energy game in general, in the sense that they achieve a win from some finite (but possibly non-optimal) energy level.

Algorithmic problems. We discuss three algorithmic problems, in increasing order of difficulty, which can be instantiated to both variants leading to six (closely related) problems.

- Determine the set of vertices with mean-payoff value $\geqslant 0$ or with finite energy-value (threshold problems).
- Determine the value of each vertex (value problem).
- Construct an optimal strategy for each player (strategy synthesis).

In the case of energy games, it is not hard to synthesise optimal strategies directly from the values of the vertices, and moreover all known algorithms for the threshold problem actually compute the values. For this reason, we will simply say *solving an energy game* for the problem of computing the values, or equivalently constructing energy-optimal strategies.

As a direct consequence of their positionality, it turns out that both threshold problems are equivalent in a strong sense: a vertex has mean-payoff value $\geqslant 0$ if and only if it has finite energy value. Current state-of-the-art algorithms for the threshold problem actually solve the energy game. The value and strategy synthesis problems for mean-payoff games are then generally solved by reducing to many instances of the threshold problem, with techniques often involving a dichotomy on N.

We will later concentrate on solving the threshold problem, generally via solving the energy game, which seem to capture the overall complexity, and state-of-the-art approaches. Moreover, energy games offer a combinatorial handle to the resolution of mean-payoff games, which turns out to be well-suited to our approach. As was previously mentioned, there is an easy reduction from (finite) parity games due to Puri [Pur95], simply by replacing each priority p with the weight $(-n)^p$, where as always n denotes the number of vertices. This gives another motivation for focusing rather on the threshold problem; significance of mean-payoff games is discussed in more length below.

2.1 From origins to state of the art

We will say that a runtime bound which does not depend[10] on the maximal absolute value N of a weight is *combinatorial*. An algorithm whose runtime is polynomial in n and N is called *pseudopolynomial*. We raise the reader's attention on the difference between quasipolynomial and pseudopolynomial algorithms: the former have runtime $n^{O(\log^c n)}$, while the latter have runtime $O((nN)^c)$.

Early work. Unlike parity games which come from logics, mean-payoff games are rooted within geometry. Early appearances of related formalisms can be traced back to the work of Gilette [Gil57] (see also rectifications made by Liggett and Lippman [LL69] to wrong claims of Gilette), who establishes existence of stationary strategies for (concurrent, stochastic) mean-payoff games as a degenerated case of Shapley's [Sha53] stochastic games.

Turn-based, non-stochastic finite mean-payoff games as above were first considered in the seminal work of Ehrenfeucht and Mycielski [EM73; EM79], who established their positionality. Their proof is short and elementary, and is based on interactions with so-called cyclic games, which are finite duration and stop as soon as a cycle is closed. Cyclic games have later been considered in [VJ00] (for parity games) and [BSV04b]. Generalisations have been studied by Karzanov and Lebedev [KL93], who were also probably the first to state that the threshold problem belongs to NP $\cap$ coNP, as a consequence of the main result of [GKK88], discussed just below. It is worth mentioning that Ehrenfeucht and Mycielski state that more direct proofs would be desirable.

A second study of mean-payoff games was given by Gurvich, Karzanov and Khachivan [GKK88], who present an algorithmic proof[11] of the existence of *ergodic potentials*, which implies a positionality proof and an algorithm for strategy synthesis. Quite notably, their approach is based on a subroutine with exponential runtime for computing the energy values (such a terminology is anachronic) via successive *potential transformations*, which is used as part of a global dichotomy. The subroutine in question has later been called the GKK algorithm and the analysis given in [GKK88] for its termination yields a combinatorial bound of $n2^n$ iterations (which was not stated explicitly, unfortunately), each of which has runtime $O(m)$. We believe that the GKK algorithm, which is attractor-based, is often overlooked in the recent literature (probably due to the low publicity of the Journal of USSR in which it was published); more details will be given in Chapter 10.

A third great introduction to mean-payoff games is given in the seminal work of Zwick and Paterson [ZP95; ZP96]. Relying on the previously established positionality, Zwick and Paterson gave direct algorithms with pseudopolynomial runtime respectively $O(n^2mN)$, $O(n^3mN)$ and

[10] Formally, this requires a model of computation able to deal with integer operations of magnitude N in constant time. We will always implicitly work in the RAM model with word size $\log N$, but abstain from the use of so-called "RAM tricks", which encode (potentially non integral) additional data on the RAM to incur further (generally logarithmic) speedup.

[11] It is also noted in [GKK88] that (non-effective) existence of ergodic potentials follow from much more general results of Moulin [Mou76] (in French), and also has an earlier proof by Parthasarathy and Raghavan [PR71].

$O(n^4 mN \log(m/n))$ for the threshold, value, and strategy synthesis problems. Perhaps equally importantly, Zwick and Paterson established a reduction from mean-payoff games to Condon's simple stochastic games [Con90] (via non-stochastic, turn-based discounted games), which also lie in NP $\cap$ coNP and have attracted a lot of attention.

We also mention a related work of Pisaruk [Pis99] who further studied generalisations of cyclical games. Pisaruk established a general pseudopolynomial algorithm in this setting, which instantiates to the GKK algorithm when the input corresponds to a mean-payoff game. In particular, Pisaruk established[12] a runtime bound of $O(n^2 mN)$ for the GKK algorithm, matching the one of Zwick and Paterson.

The study of mean-payoff games as a simple graph-based problem lying in NP $\cap$ coNP and not known to be in P was first advocated by Zwick and Paterson [ZP96], who have considerably participated in popularising the problem. Despite more than two decades of considerable efforts, no substantial progress has been made on this front.

Strategy improvements. It was known from the aforementioned works of Zwick and Paterson [ZP95], Puri [Pur95], and Ludwig [Lud95] that mean-payoff games can be reduced to discounted or simple stochastic games over which strategy improvements can be performed, and even randomised strategy improvements with subexponentially-many iterations. However it was not until almost a decade later[13] that specific strategy improvement algorithms were given for mean-payoff games by Björklund, Sandberg and Vorobyov [BSV04a]. Their contribution has two aspects.

First, they provided a framework for strategy improvements applied directly to mean-payoff games. The obtained formalism is conceptually simpler than that of Vöge and Jurszinski [Vög00; VJ00], and based on longest-shortest paths, which are in essence quite similar to energy games, but they require so-called *retreats* or *admissible* strategies. Similar technicalities were later used in Schewe's optimal strategy improvement [Sch08]. These will be discussed in Chapters 9 and 10.

Second, Björklund, Sandberg and Vorobyov gave a first subexponential algorithm in the form of a randomised switching policy in their new framework. The obtained complexity for the threshold problem is min $\big(O(mn^2N), 2^{O(\sqrt{n \log n})}\big)$; note that the first bound match the previous pseudopolynomial ones. By reduction (dichotomy) they also obtain a procedure for the value problem with complexity $O(n^3 mN \log(nN), \log(N)2^{O(\sqrt{n}\log n)})$, again roughly matching the one of Zwick and Paterson, up to a logarithmic factor. The randomised subexponential bound for the value problem was later improved to by Andersson and Vorobyov [AV06] to roughly $2^{O(\sqrt{n \log(m/\sqrt{n})})}$ (strongly subexponential), which is the best currently available combinatorial bound (if one includes randomised algorithms).

Energy games. Energy games were first studied by Chakrabarti, de Alfaro, Henzinger and Stoelinga [CAH+03] in the context of resource interfaces and their analysis. Among other results for several variations, a simple fixpoint algorithm with runtime $O(n^3 N)$ was presented for computing the energy values, in the setting where weights label the vertices of the game. Positionality of energy games is not formally discussed.

Bouyer, Fahrenberg, Larsen, Markey and Srba [BFL+08] were the first to establish positionality of energy games for both players (over finite games), and to present a connection with mean-payoff

[12] The proof of Theorem 3 therein establishes $O(n^2 N)$ iterations (for mean-payoff games we have $T_{\mathcal{F}} = O(1)$ in Pisaruk's notations) for his algorithm, which instantiates to the GKK algorithm for mean-payoff games. Moreover, each iteration has runtime $O(m)$ in this case.

[13] This is not entirely true: a strategy improvement algorithm was presented earlier by Cochet-Terrasson, Gaubert and Gunawardena [CTGG99] (see also [GG98]) in the closely related context of min-max functions. However the algorithm is presented in a completely geometric fashion; it is not clear how to derive a combinatorial framework from their work.

games (their intent was to provide hardness of energy games, for which they proved the NP $\cap$ coNP upper bound). However the precise threshold problems considered in [BFL+08] for both formalisms are not "aligned" as they are here (mean-payoff value $\geqslant 0$ versus energy value finite), and thus the translation requires a logspace reduction.

Brim, Chaloupka, Doyen, Gentilini and Raskin [BCD+11] later realised (see Theorem 3 therein) that the positionality of energy games (established by [BFL+08]) implies a direct equivalence between both threshold problems. Based on this observation, they gave a natural value iteration algorithm for solving energy games in time $O(mnN)$, which improves on all previous pseudopolynomial bounds for the threshold problem, and which we will refer to as the BCDGR algorithm. It is based on a fixpoint formulation of energy values (not unlike those of [CAH+03] and [BFL+08]), and an improved Kleene iteration (not unlike that of [Jur00]).

By a dichotomy (not unlike those proposed in [GKK88] or [BSV04a]) the authors of [BCD+11] extended their approach to solve the value and strategy synthesis problems, obtaining the pseudopolynomial bounds $O(mn^2N \log(nN))$. A subtler reduction to energy games (which also makes repeated use of the BCDGR algorithm) was later presented by Comin and Rizzi [CR15; CR17], and improved in [CR16], which establishes the state-of-the-art pseudopolynomial bound of $O(n^2mN)$ for the value and strategy synthesis problems, removing the $log(nN)$ factor from the solution of [BCD+11].

Recently, a novel deterministic algorithm for solving energy games was presented by Dorfman, Kaplan and Zwick [DKZ19], which implements a scaling technique on top of an acceleration of the BCDGR algorithm. In the full version of the paper[14] the need for scaling was removed, and the accelerating subroutine simplified, leading to a runtime of $O(\min(mnN, m2^{n/2}))$. This is an improvement on the BCDGR algorithm since it adds a combinatorial upper bound, which currently holds the state of the art for deterministic algorithms. Previously, the best deterministic combinatorial bound was $O(mn2^n)$ which follows from the analysis of [GKK88] for their algorithm, and was also obtained[15] by Lifshits and Pavlov [LP07].

Similarities between the (attractor-based) GKK algorithm and the (accelerated value iteration) algorithm of Dorfman, Kaplan and Zwick will be further discussed in Chapter 10. An extension of the approach to discounted games was recently given by Kozachinskiy [Koz21b], establishing a deterministic combinatorial $n^{O(1)}(2+\sqrt{2})^n$ bound, whereas no deterministic algorithm with runtime $2^{o(n \log n)}$ was previously known for discounted games with arbitrary discounts.

Practicality. We are not aware of any systematic comparison of the different algorithms for mean-payoff games in the literature, or of benchmarks regrouping games arising from practical applications (other than translating benchmark from parity games, which are inherently non-quantitative thus not fit for capturing practicality of mean-payoff games). Our empirical experiments suggest that, at least over randomly generated games, strategy improvements are more scalable than the GKK algorithm, which itself is much more scalable than the BCDGR algorithm. This last point is not surprising, value iterations are well-known to frequently display their worst-case complexity. We do not exclude (but would be surprised by) the possibility that efficient implementations of the attractor-based GKK algorithm can challenge strategy improvements in practice (as is the case for Zielonka's algorithm for parity game).

Similar observations have been made in the literature, see for instance [Sch08] or [DG06]. We mention also a paper of Meyer and Luttenberger [ML16], which presents an implementation of

[14] It has not appeared yet, but is available on Dorfman's webpage.

[15] A $O(mn2^n \log N)$ bound is given for the mean-payoff value problem, but it is obtained by reduction to a procedure (see Section 4 therein) which solves the energy game in time $O(mn2^n)$, matching the GKK algorithm. Lifshits and Pavlov seem unaware of the details of the GKK algorithm which they qualify as "pseudopolynomial", and they refer to Gallai [Gal58] for the use of potentials.

the optimal strategy improvement [Sch08; Lut08] on graphical units, and reports on considerable speed-up over large instances.

The tropical approach. An increasingly rich body of work initiated by Allamigeon, Gaubert and co-authors, studies interactions between mean-payoff games, tropical geometry and techniques from non-archimedean optimisation. A few pointers to the recent literature include [AGS18; AGK+18; AGS20; AGQ+21] and Loho's PhD thesis [Loh17]. One of these approaches, from Allamigeon, Benchimol, Gaubert and Joswig [ABG+13], is based on rephrasing the threshold problem for the mean-payoff valuation as a conjunction of polynomially many linear inequations in the tropical semiring $(\mathbb{R} \cup \{-\infty\}, \max, +)$, and then importing ("tropicalising") known methods and results from linear programming.

The same authors established in [ABG+13] that combinatorial switching rules can be tropicalised and in particular a combinatorial simplex algorithm with strongly polynomial complexity would imply a strongly polynomial algorithm for mean-payoff games. Another striking outcome of this technique, presented in [ABG14], consists in tropicalising the shadow-vertex pivoting rule from Adler, Karp and Shamir [AKS87], leading to a deterministic algorithm for mean-payoff games which exhibits polynomial time in average, as long as the game is drawn from a flip-invariant distribution.

Most results obtained in this line apply to the more general *stochastic* mean-payoff games. We see this as indication that their geometric approach is essentially orthogonal to the one we will adopt: energy games appear to be somewhat irreconcilable with the stochastic setting in which there seems to be no reasonable definition for energy values.

2.2 Significance, applicability and extensions

Theoretical motivations. Several theoretical motivations for studying mean-payoff games were described above: like parity games, which they generalise, mean-payoff games belong to NP ∩ coNP, and as observed by Jurdziński [Jur98], even to UP ∩ coUP since they can be reduced to discounted games which admit canonical strategies. We have also seen that mean-payoff games are reducible to simple stochastic games which have attracted a lot of attention and also belong to NP∩coNP, and its close connection to linear tropical optimisation provides yet another motivation. Last, reductions (sometimes with large blow-up) were often described from games with more exotic winning conditions to mean-payoff games – a notable example being the impressive (and involved) chain of reductions established by Colcombet, Jurdziński, Lazić and Schmitz [CJL+17] – which gives another incentive to find efficient solutions to mean-payoff games. Although mean-payoff games find practical applications in verification (see for instance [DG06]), they are most notably relevant in the context of reactive synthesis.

Quantitative synthesis. The introduction of energy games by Chakrabati, de Alfaro, Henziger and Stoelinga [CAH+03] was made in the context of resource interfaces, which is related to reactive controller synthesis. It already appeared in this work that real-life applications of quantitative specifications may require more expressive formalisms than so-called *pure energy*. Indeed reward energy interfaces are introduced, modeled by conjunctions of an energy and a Büchi objective.

Integrating quantitative constraints in reactive synthesis specifications has very often been proposed. We are not aware however of any framework, implemented or not, that employs a *pure* mean-payoff (or energy) solver as its end-component. The focus in this context is really on deriving adequately expressive quantitative objectives, which often generalise those above. We briefly survey a few of them, a more thorough (yet slightly outdated) overview can be found in Randour's thesis [Ran14], which includes stochastic generalisations, omitted here but often relevant in applications.

Mean-payoff parity games and the like. Mean-payoff parity games are given by a conjunction of a mean-payoff and a parity objective and were introduced by Chatterjee, Henziger and Jurdziński [CHJ05] who established existence of optimal strategies, infinite memory requirement for the conjunctive player, and gave an algorithm reducing to n^d resolutions of parity and mean-payoff games. In most well-studied extensions (see also below), it turns out that mean-payoff and energy objectives no longer coincide. Energy parity games were later studied by Chatterjee and Doyen [CD10; CD12], who proved positionality for the disjunctive player, membership in NP $\cap$ coNP[16] and polynomial time equivalence with mean-payoff parity games (and thus their membership in NP $\cap$ coNP, which was unknown), and gave an algorithm with similar complexity.

Bouyer, Markey, Olschewski and Ummels [BMO+11] later applied mean-payoff parity games for the reactive synthesis of *permissive* strategies via *penalties* (see also [BDM+09]). In this modular quantitative approach, non-deterministic controller strategies are sought, allowing for as many different desirable behaviours as possible; the obtained decision problem can be formulated as a mean-payoff parity game. On the way, the authors of [BMO+11] propose a second study of mean-payoff parity games, establishing their positional determinacy (for the disjunctive player), giving easier proofs for existence of optimal strategies, and a more efficient and conceptually simpler algorithm. Mean-payoff parity games have been integrated in several other reactive synthesis frameworks which combine qualitative (parity) objectives for functionality, and quantitative (mean-payoff) objectives for performance or robustness (see for instance [BBR13] or [BCG+14], which include in-depth discussions about several related works).

Other related games have been studied, such as parity games with costs, by Fijalkow and Zimmermann [FZ14], generalised to parity games with weights by Schewe, Weinert and Zimmermann [SWZ19], which they also proved to be polynomial-time equivalent to energy parity games, and therefore to mean-payoff parity games. Quite notably, all algorithms presented so far are extensions of Zielonka's recursive attractor-based algorithm, and therefore inherit the exponential dependency in d. We mention also the work of Chatterjee, Henzinger and Svozil [CHS17] who provided a $O(mnN)$ algorithm for the case where there are only two priorities, extending the state-of-the-art bound for pure mean-payoff games to this case, which is often relevant in applications.

The state of the art for mean-payoff parity games is due to Daviaud, Jurdziński and Lazić [DJL18], who generalised the quasipolynomial value iteration from parity to parity mean-payoff games, establishing a pseudo-quasipolynomial bound of order

$$mn^2N\binom{d/2+\log n}{d/2},$$

which matches the one for parity games up to an additional multiplicative nN. This algorithm will be further discussed in Chapter 7. Unfortunately, the state-of-the-art algorithm inherits the impracticability of value iteration approaches, which are known to frequently exhibit their worst-case runtime. We are not aware of implementations of (full) mean-payoff parity solvers; actually, solving large scale mean-payoff parity games even with a small fixed number (say, five) of priorities appears to be elusive with the currently known techniques.

Multi mean-payoff games. A rich line of research concerns mean-payoff and energy games with multiple dimensions, introduced by Chatterjee, Doyen, Henzinger and Raskin [CDH+10]. This

[16] This is surprising since the conjunctive player may require exponential memory. Such results (as is the case here) are generally based on non-trivial decompositions for optimal strategies.

initial work establishes existence of optimal strategies in both cases, and even finite memory strategies for multi energy games, and proves that when restricted to finite memory strategies, both formalisms coincide. Among other complexity results, they also established coNP-completeness for multi energy games when the dimension is not fixed, and even when $N = 1$. The multidimensional formalism was applied in reactive synthesis by Cerný, Gopi, Henzinger, Radhakrishna and Totla [CGH+12] for systems modeling the evolution of several different resources, allowing to find tradeoffs between possibly incompatible specifications.

Chatterjee and Volner [CV13] proposed a hyperplane separation technique for the multi mean-payoff problem, establishing an upper bound in $m(knN)^{O(k^2)}$, which is pseudopolynomial for fixed dimension k. A similar technique was later employed by Jurdziński, Lazić and Schmitz [JLS15] to obtain a similar bound $O(nN)^{O(k^4)}$ for multi energy games, consequently settling several open problems for the closely related setting of games played on vector addition systems with states. Together with Colcombet in [CJL+17], the same authors gained considerable insight by formulating the technique in game theoretic terms, moreover allowing to encompass conjunctions with parity games (introduced by Abdulla, Mayr, Sangnier and Sproston [AMS+13]), with state-of-the-art complexity $O((nN)^{(d+p)^3 \log(d+p)})$. A tractable variant will be discussed in Chapter 8.

Other variants and extensions. Over the last decade or so, numerous other extensions and variants have been considered in the context of reactive synthesis; we (non-exhaustively) give a few pointers without discussing further details. Arbitrary boolean combinations of mean-payoff games were shown to be undecidable in [Vel15]. Energy games with bounds were studied in [FJL+11] and [JLR13]. Average energy games were introduced in [BMR+15], while bounded and multidimensional extensions are studied in [BHM+17]. Perhaps surprisingly, conjunctions of an energy and a mean-payoff objective had not come under scrutiny until very recently [BHR+19]. These can be seen as one-letter pushdown mean-payoff games, which are undecidable in general [CV12]. Pushdown (single and multidimensional) energy games are studied in [AAH+14].

Applications in reactive synthesis have recently called for the development of a number of (often quantitative) game-theoretic frameworks. The two most relevant questions are then

- structural: how simple are optimal strategies (positional, finite memory, other structural insight)? or
- algorithmic: can the winner be efficiently decided? better, can optimal strategies be synthesised?

This motivates the study of *modular* and *generic* theoretical tools for tackling the two questions.

3 Contributions and organisation of the thesis

We introduce *monotonic graphs* which are totally ordered graphs in which edge relations are monotonic, and advocate their use for the study of (finite and infinite) games which are *positionally determined for Eve*[17]. These simple and natural objects have made numerous more or less explicit apparitions in the literature, which will be discussed throughout.

Monotonic graphs allow to establish game-theoretic properties and algorithmically synthesise strategies as long as they realise a *graph-theoretic* property, namely, universality with respect to the

[17] In this thesis, "positionality" always refers to positionality *for Eve*, which is the relevant property in synthesis, sometimes called "half positionality". To refer to "positionality for both players", we will speak of "bi-positionality". These notions are formally introduced in the preliminaries.

condition under study. Our work builds on the one of Colcombet and Fijalkow [CF18; CF19] (see also [CFG+21], currently under review) who identified graph-universality as a notion which on one hand captures recent advances (universal trees) when instantiated to parity games, and on the other lends itself to further generalisations. After a preliminary part introducing all needed notions (which are mostly standard) and classical results, the thesis is organised in three parts, further split in a total of 11 chapters. Different parts are based on collaborations with different colleagues, who will be acknowledged in corresponding chapters.

Part I. We start by introducing monotonic graphs, and generalising the framework proposed by Colcombet and Fijalkow by dropping the assumption of prefix-independence and moreover introducing quantitative behaviours. As one of our main contributions, we establish in this general context that positionality over arbitrary arenas is equivalent to the existence of universal well-ordered monotonic graphs. This is the first characterization for (half) positionality.

On the way, we obtain a novel closure property: prefix-independent positional objectives are closed under lexicographical products. As far as we are aware, no similar general closure property is known. We also use monotonic graphs as tools to obtain various positionality results (most of which are known). We include comparisons with related works, and discussions about structural perspectives opened by our approach, as well as its limitations.

Part II. We then switch our focus from structural to algorithmic questions, for which monotonic graphs reduce solving games to computing a (least) fixpoint. The second part revolves around computing the fixpoint simply by Kleene iteration, which corresponds to a (generic) *value iteration* algorithm that we also show to be roughly equivalent in general to the strong separation approach of Bojańczyk and Czerwiński [BC18].

We then present a few case studies. We start with parity games, for which *saturated* (monotonic[18]) graphs correspond with trees and therefore universal (monotonic) graphs can be embedded in universal trees. For completeness, we also present the quasipolynomial construction of Jurdziński and Lazić [JL17] and the almost matching lower bound of Fijalkow [Fij18] over universal trees.

We go on to study threshold mean-payoff games, for which a natural monotonic graph directly follows from the connection with energy games, and the obtained value iteration algorithm coincides with the (state-of-the-art) BCDGR algorithm. We provide additional upper and lower bounds in this case, essentially concluding that value iterations are unlikely to yield more efficient algorithms.

We then turn to mean-payoff parity and multi mean-payoff games. For the former, we give an alternative presentation of the value iteration algorithm of Daviaud, Jurdziński and Lazić [DJL18] as a universal monotonic graph; the universality proof required here turns out to be non-trivial and (we believe) interesting in its own right. For multi mean-payoff games of dimension d, we only focus on the tractable (much easier) variant where the lim sup semantic is used, and show how to combine monotonic graphs for parity and energy games to obtain a value iteration with runtime $O(mdn\log(n)N)$, essentially gaining a factor n over [VCD+15].

Part III. In the third part, we explore the possibility of computing the desired fixpoint by other means than Kleene iteration. We plead that (universal) monotonic graphs provide the right framework for *strategy improvements* by showing that the necessary condition of being positional for Adam is actually *sufficient* for running such an algorithm, for valuations induced by monotonic graphs. Prior to our work, different abstract frameworks for performing strategy improvements subject to different sufficient conditions have been put forth, but as far as we know such a characterisation is

[18] It follows from generic results from Part I that saturated graphs with respect to positional conditions are monotonic.

novel. We also discuss implications for parity and mean-payoff games, relations with existing work, and perspectives.

So far, all algorithms were asymmetric: a player is chosen arbitrarily and the corresponding fixpoint is computed. Our two final chapters propose to study *symmetric attractor-based* algorithms, respectively for mean-payoff and parity games.

For mean-payoff games, it appears that there are only two (natural) monotonic graphs, corresponding to the two players, to the usual order over $\mathbb{Z}$ and its dual, or to the energy valuation and its dual. Therefore algorithms based on monotonic graphs are intrinsically related with *potentials*, which we attribute in this context to Gurvich, Karzanov and Khachivan [GKK88] and were often rediscovered thereafter.

We study *simple* mean-payoff games, which exclude zero cycles; this assumption can be lifted in general at the cost of multiplying N by n. First, we show that for simple mean-payoff games, the (attractor-based) GKK algorithm admits a completely symmetric presentation, and we propose a symmetric analysis revealing a novel upper bound of $N+E^+ +E^- +1 = O(nN)$ on the number of iterations (each has runtime $O(m)$), where E^+ and E^- denote the maximal finite energy and dual-energy values. For simple mean-payoff games, this is a substantial improvement on the BCDGR algorithm, in which the constant nN is hardcoded, whereas $N + E^+ + E^-$ may be much smaller. We also re-establish the recent combinatorial $O(m2^{n/2})$ bound of Dorfman, Kaplan and Zwick by adapting their method to the GKK algorithm.

Second, guided by insight gained in our study of strategy improvements, we propose a simplification of Schewe's [Sch08] and Luttenberger's [Lut08] optimal scheme, which is a crucial component in the successful framework of STRIX [LMS20]. The obtained presentation naturally suggests a symmetric variant[19], who appears to be even more practical, but whose termination eludes our current toolset.

Different monotonic graphs for Eve (or Adam) for parity games can be naturally interpreted as different ways of *measuring* occurrences of odd (or even) priorities[20]. In this regard, rather than interpreting a single monotonic graph (namely $\mathbb{Z}$) from the point of view of both players as we did for mean-payoff games, it seems more adequate here to *interleave* two monotonic graphs, one for each player. Our final chapter explores this idea, which reveals a surprising (and fascinating, we believe) connection between value iteration and attractor-based algorithms for parity games.

Recall that attractor-based algorithms repeatedly discard considerable amounts of information (see step 2 in Figure 2), whereas information in value-iterations is aggregated completely monotonically hence it is never lost. We thus propose to use monotonic graphs as adequate data structures for improving attractor-based algorithms.

More precisely, we first show that the recent universal attractor-decomposition algorithm of Jurdziński and Morvan [JM20], which is parameterised by two trees T^{odd} and T^{even}, can be simulated simply by (independently) running*parallel* value iterations (one for each player, in each of the trees). In particular, this gives an alternative correctness proof for their algorithm.

Second, we present a natural *acceleration* mechanism in this setting, which uses information computed in each monotonic graph to speedup the other iteration. This allows us to define a new generic class of iterative algorithms, which encompasses all known quasipolynomial algorithms so far, but also Zielonka's algorithm, and variants of Zielonka's algorithm (or other attractor-based

[19] The algorithm in question is completely different from the symmetric strategy improvement of [STV15].

[20] We recall to the reader familiar with universal trees that these correspond to *saturated* monotonic graphs for the parity condition. Monotonic graphs in general may correspond to other ways of counting occurrences, and non-saturated monotonic graphs are relevant here.

algorithms) that do not discard information. The analysis of the obtained class of algorithms appears to be both exciting and challenging; we also propose a few directions for future work on this front.

Preliminaries

Notational conventions. For clarity, we often use the notation $a_{b,c}$ for $(a_b)_c$, and also apply this to superscripts, for instance $a^{b,c}$ stands for $(a^b)^c$. We also make use of (sometimes intelligent, but always reasonable) completion, for instance

$$x_0 \xrightarrow{a} x_1 \xrightarrow{b} x_2 \xrightarrow{a} x_3 \xrightarrow{b} x_4 \xrightarrow{a} \dots \xrightarrow{b} x_{2k},$$

should be understood as "$k \geqslant 0$ is such that for all $i \leqslant k-1$, $x_{2i} \xrightarrow{a} x_{2i+1}$ and $x_{2i+1} \xrightarrow{b} x_{2i+2}$".

1 Orders and graphs

Relations. A *relation* over X is a subset of $X \times X$. Given a relation $R \subseteq X \times X$ and $x, x' \in X$ we use $x \mathrel{R} x'$ to denote $(x, x') \in R$, and as is standard we extend this notation to sequences of elements, for instance $x \mathrel{R} x' \mathrel{R'} x''$ means $(x, x') \in R$ and $(x', x'') \in R'$. We will essentially consider two types of relations: various notions of orders on one hand, and edge relations on the other.

For edges, we will use the visual notation $\rightarrow$, and as our edges will generally be coloured by $c \in C$, we will therefore write $x \xrightarrow{c} x'$. Since we will very often consider orders and edges over the same underlying X, we align our conventions and therefore use $\geqslant$ to define orders, despite the well-established tendency of preferring $\leqslant$ by default. We also generally favor the use of non-strict orders.

1.1 Orders, lattices and well-orders

We start with terminology and notations relative to orders; graphs and edges are dealt with in the following section. Although these are omnipresent in our work, we will only require very basic order theory, and refer to wikipedia.

Orders. A relation $\geqslant$ over X is said to be (below, x, x', x'' range over X and are quantified universally)

- *reflexive* if $x \geqslant x$;
- *transitive* if $x \geqslant x' \geqslant x''$ implies $x \geqslant x''$;
- *a preorder* if it is both reflexive and transitive;
- *antisymmetric* if $x \geqslant x'$ and $x' \geqslant x$ imply $x = x'$;

- *a (partial) order* if it is preorder which is moreover antisymmetric;
- *total* if either $x \geqslant x'$ or $x' \geqslant x$;
- *a linear order* if it is an order which is total;
- *symmetric* if $x \geqslant x'$ implies $x' \geqslant x$ (orders are not usually symmetric);
- *an equivalence* if it is reflexive, symmetric and transitive.

Given such a relation, we use $\leqslant$ to denote its *dual*, defined by $x \leqslant x'$ if and only if $x' \geqslant x$, and $<$ and $>$ to denote respectively the negations of $\geqslant$ and of its dual.

We raise the reader's attention on the fact that

$$x > x' \qquad \Longleftrightarrow \qquad x \geqslant x' \text{ and } x \neq x'$$

does not hold in every case, although it does holds for linear orders. We will often consider total preorders, for which the above is not true.

A total preorder $\geqslant$ induces an equivalence $\equiv$ over the same set X given by

$$x \equiv x' \qquad \Longleftrightarrow \qquad x \geqslant x' \text{ and } x' \geqslant x.$$

Moreover $\geqslant$ naturally induces a linear order over equivalence classes[21] of $\equiv$, which as is standard, we also denote by $\geqslant$.

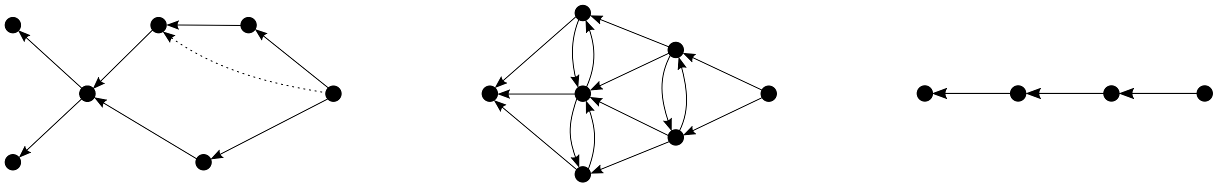


Figure 4: From left to right, a (partial) order, a total preorder and a linear order. An arrow $x \to x'$ corresponds to $x \geqslant x'$; for clarity, we do not depict arrows which follow from transitivity, such as the one which is dashed. The equivalence classes of the preorder are given by vertices which are aligned vertically, and these are linearly ordered as on the right. The partial order on the left is not complete (see below), since the two leftmost elements do not have an infimum.

Given a linear order $\geqslant$ over X, we let $[a, b]$ denote the *interval* between a and b, which is the set of elements $x \in X$ satisfying $b \geqslant x \geqslant a$. We use parentheses to exclude bounds, for instance $[a, b)$ denotes the set of x's such that $b > x \geqslant a$. When X is not clear from context, we may add it as a subscript for clarity, for instance $[3, 4]_{\mathbb{Q}}$ denotes the set of rational numbers between 3 and 4.

Complete lattices. Fix a partial order $\geqslant$ over a set L and a subset S of L. An element $\ell \in L$ is an *upper bound* of S if $\ell \geqslant s$ for all $s \in S$, and it is a *supremum* (or *least upper bound*) if any upper bound ℓ' satisfies $\ell' \geqslant \ell$.

By antisymmetry there cannot be more than one supremum to a given S. A supremum $\ell \in L$ of S is called a *maximum* (or *greatest element*) of S if it belongs to S. If it exists, the maximum of S is then the unique element $\ell \in S$ satisfying $\ell \geqslant s$ for all $s \in S$. We define *lower bounds*, *infima* (or *greatest lower bounds*) and *minima* (or *least elements*) dually.

[21] Since these will appear only very sporadically, we do not formally define quotients and equivalence classes.

A *lattice* is a partially ordered set L in which every finite set (or equivalently, every pair of elements) S admits a supremum and an infimum in L. In this case they are unique, and respectively denoted $\sup S$ and $\inf S$. A *complete lattice* is a lattice in which the same property holds even for infinite sets. Note that a complete lattice L has a maximum and a minimum, namely $\sup L \in L$ and $\inf L \in L$, which we generally denote $\top$ and $\bot$ respectively. We say that an order $\geqslant$ over L is *complete* if it equips L with a complete lattice structure.

Examples of complete lattices. The sets $\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}$ of natural, relative, rational and real numbers, are all linearly ordered by the usual order $\geqslant$. Their finite sets admit minima and maxima (which are very special cases of suprema and infima) therefore the three are lattices.

None of $\mathbb{N}, \mathbb{Z}, \mathbb{Q}$ or $\mathbb{R}$ are complete lattices, since they do not admit maxima, but $\mathbb{N} \cup \{\infty\}$, $\mathbb{Z} \cup \{-\infty, +\infty\}$ and $\mathbb{R} \cup \{-\infty, \infty\}$ are, the later being equivalent (in terms of orders) to the closed interval $[-1, 1]_{\mathbb{R}}$, which is also a complete lattice. Infinite intervals of $\mathbb{Q} \cup \{-\infty, +\infty\}$ such as $[-1, 1]_{\mathbb{Q}}$ or $[0, \infty]_{\mathbb{Q}}$ are not complete as lattices, since they contain subsets whose suprema belong to $\mathbb{R}\backslash\mathbb{Q}$.

Other fundamental examples of complete lattices include *powerset lattices*, which are those of the form $L = \mathcal{P}(X)$ where X is an arbitrary set, and where suprema and infima are given by unions and intersections. A standard equivalent way of seeing $\mathcal{P}(X)$ is as the set 2^X of functions from X to $2 = \{0, 1\}$, and in this case unions and intersections correspond to pointwise maxima and minima of functions, with respect to the natural order on the pair. Note that the pair 2 is itself a complete lattice, and actually replacing it above by an arbitrary complete lattice L yields a complete lattice L^X, (partially) ordered pointwise.

Stated differently if L is a complete lattice and X is an arbitrary set then L^X is a complete lattice. Note that even the if order over L is total, the induced pointwise order over L^X is not in general (unless X is a singleton). This observation is particularly relevant to our work in which a central role is played by lattices L^X of functions into complete lattices L which are linearly ordered.

Fixpoints and the Knaster-Tarski theorem. A function $f : X \to X'$ between partially ordered set is *monotonic* (or *order-preserving*) if it preserves the order, formally

$$x \geqslant x' \text{ in } X \qquad \Longrightarrow \qquad f(x) \geqslant f(x') \text{ in } X'.$$

An *operator* is a function $f : X \to X$ with matching domain and codomain. When X is partially ordered, an element $x \in X$ is a *prefixpoint* of an operator f if $x \geqslant f(x)$, a *postfixpoint* if $f(x) \geqslant x$, and a *fixpoint* if $f(x) = x$. The following theorem was first established by Knaster and Tarski [KT28] for powerset lattices and later extended to general complete lattices by Tarski [Tar55]. It is very well-known by logicians and finds countless applications all over computer science.

Theorem 1 (Knaster-Tarski theorem)

The set of fixpoints of a monotone operator over a complete lattice is itself a complete lattice. In particular, it has a minimal element, which moreover coincides with its least prefixpoint.

Well-orders and ordinals. A linear order $\geqslant$ over X is a *well-order* if all non-empty subsets $S \subseteq X$ have a minimum. Equivalently, any non-increasing infinite sequence of elements $x_0 \geqslant x_1 \geqslant \dots$ is stationary. Well-ordered sets are closely related to *ordinal numbers*. Formally, ordinal numbers are defined to be those sets over which set membership defines a strict well-order.

What is important to us is that ordinals characterise well-orders, in the sense that any well-ordered set is order-isomorphic to an ordinal. These should be seen as refinements of cardinals:

cardinals represent sets up to bijection, whereas ordinals represent well-ordered sets[22] up to order-preserving bijections.

Von Neumann's notation of ordinals is given by $\lambda = [0, \lambda)$, for instance $2 = \{0, 1\}$ where $1 = \{0\}$ and $0 = \varnothing$. As is standard, we use Greek letters to denote ordinals, and ω denotes the first infinite ordinal $\omega = \{0, 1, 2, \ldots\} = [0, \omega)$ which is equipotent to $\mathbb{N}$, and order-isomorphic to $\mathbb{N}$ when it is equipped with the usual order. We then have $\omega + 1 = [0, \omega + 1) = [0, \omega] = \omega \cup \{\omega\}$, which is order-isomorphic to $\mathbb{N} \cup \{\infty\}$, and followed by many subsequent ordinals, $\omega + 2, \omega + 3$ and so on. From now on, we use the ordinal notation ω to denote the set of natural numbers, and also sometimes for finite ordinals (or natural numbers), for instance $n = [0, n - 1]$.

Well-orders are particularly useful in that they allow to generalise the principle of induction from the set ω of natural numbers, to any set equipped with a well-order (equivalently, to any ordinal). In this context, we speak of *transfinite induction*, which stipulates that

a property $P(\alpha)$ which holds whenever $P(\beta)$ holds for all ordinals $\beta < \alpha$,
holds for all ordinals.

Stated differently, it suffices to show that a property passes on to α when it holds for all smaller ordinals to show that it holds for all ordinals.

Well-orders which are also complete lattices will later play an important role; observe that these correspond to ordinals which admit a maximum (often called *non-limit ordinals*) such as $\omega + 1$.

Words and prefixes. A *finite word* $w = x_0 x_1 \ldots x_{n-1}$ over a set X is a finite sequence of elements of X, and an *infinite word* $w = x_0 x_1 \ldots$ over X is an infinite sequence of elements of X. The *length* $|w|$ of a finite $w = x_0 x_1 \ldots x_{n-1}$ is its number n of elements, and the length of an infinite word is ω. When appearing in words over X, elements of X are referred to as *letters*, and the *number of occurrences* of a given letter x in a finite or infinite word w is denoted by $|w|_x \leqslant \omega$.

For $\lambda \leqslant \omega$ we use X^λ to denote the set of words over X of length λ, in particular X^ω denotes the set of infinite words, and likewise $X^{\leqslant \lambda}$ denotes the set of words of length $\leqslant \lambda$. Finally, X^* denotes the set of finite words. Note that we have $X^{\leqslant \omega} = X^* \cup X^\omega$.

Finite words can be *concatenated* by putting them next to one another, and concatenation is denoted multiplicatively; for instance if $w = x_0 \ldots x_{n-1}$ and $w' = x'_0 \ldots x'_{n'-1}$ then ww' is defined to be $x_0 \ldots x_{n-1} x'_0 \ldots x'_{n'-1}$, and it has length $|ww'| = n + n' = |w| + |w'|$. The concatenation ww' is also defined when w' is an infinite word, in which case the concatenation is infinite, and lengths behave like cardinals (hence the notation): $|ww'| = n + \omega = \omega = |w| + |w'|$. The concatenation ww' is not defined when w is an infinite word (we will not manipulate ordinal words). On some rare occasions we use a dot as in $w \cdot w'$ to improve readability. If w is finite, we let $w^\omega = ww \ldots$ be the infinite word obtained by concatenating w with itself ω times.

A *prefix* u of a finite or infinite word $w \in X^{\leqslant \omega}$ is a finite word such that $w = uw'$ for some $w' \in X^{\leqslant \omega}$. In words, u is a prefix of w if w starts with u. We use ε to denote the empty word, which is a prefix of all words. Given a finite or infinite word $w = x_0 x_1 \cdots \in X^{\leqslant \omega}$ and a natural number $n \leqslant |w|$, we let $w_{<n} = x_0 x_1 \ldots x_{n-1}$ be the unique prefix of w of length n. Note that $w_{<0} = \varepsilon$ and $w_{<1} = x_0$.

1.2 Edge-coloured graphs

We now introduce terminology relative to graphs. Graphs are based on edge relations, which are seen with a very different eye than the relations (orders) discussed so far. First, edge relations

[22] We work under the axiom of choice, which is well-known to be equivalent to the well-ordering principle, stating that any set can be well-ordered.

are arbitrary relations, no property (such as reflexivity or antisymmetry) is imposed. Second, we consider many edge relations *at the same time* which occur over the same set V, and correspond to different colours $c \in C$. Third, our focus is now on combining edges together to form paths, for instance $v \xrightarrow{c} v' \xrightarrow{c'} v''$ with different colourations. Such combinations were rendered trivial for order relations because of the transitivity assumption.

Pregraphs, sinks, graphs. We fix a set C of colours. A C*-pregraph* G over V is a family of relations over V indexed by C. It is given by a subset of $V \times C \times V$ whose elements we call *edges* and denote by $e = v \xrightarrow{c} v'$ with $v, v' \in V$ and $c \in C$. We say that c is the colour of the edge e, and in this case we say that e is a c-edge. The *size* of a pregraph G is the cardinality of $|V|$ which we usually denote by n or n_G when it is finite. We also use m or m_G to denote the cardinality of the set of edges of G. We say that G is a *finite pregraph* if *both* n_G and m_G are finite.

We call V the *set of vertices* of G. If $v \xrightarrow{c} v'$ then v is a c*-predecessor* of v' and v' is a c*-successor* of v. An edge of the form $v \xrightarrow{c} v$ is called a c*-loop around* v. We sometimes say that v and v' are the *endpoints* of an edge $e = v \xrightarrow{c} v'$ and also say that v and v' are *adjacent* to e.

Edges of the form $v \xrightarrow{c} v'$ are called *outgoing edges* from v and we let $\mathrm{Out}(v)$ denote the set of such edges. The *degree* of a vertex v is the cardinality of $\mathrm{Out}(v)$. Note that it is bounded by $|C||V|$. We say that a pregraph has *finite degree* is all vertices have finite (possibly unbounded) degree, and that it has *bounded degree* if there is a uniform finite bound on the degree of its vertices. We say that a pregraph is *finite* if it has finitely many vertices *and* finitely many edges (equivalently, it is finite and has finite degree).

A vertex v of degree 0 is called a *sink*. A C*-graph* is a C-pregraph which has no sink. Stated differently, in a graph, all vertices have a successor. Note that in a finite C-graph G we have $m_G \geqslant n_G$ in general. This terminology is a bit unusual, but well fitted to the study of infinite duration games. We simply say *graph* and *pregraph* when C is clear from context.

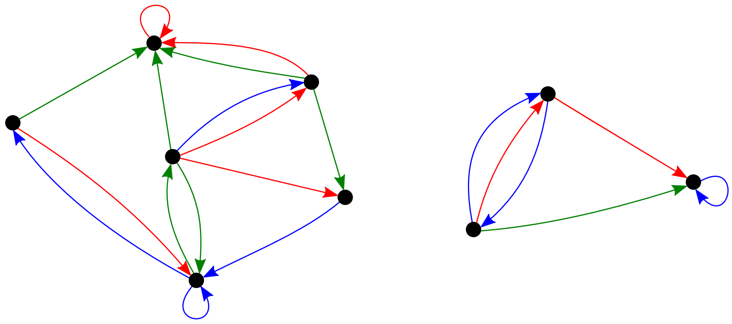


Figure 5: A finite C-graph with $C = \{\text{red}, \text{blue}, \text{green}\}$.

A *subpregraph* G' of a pregraph G is a pregraph over a subset V' of V such that all edges in G' belong to G. Given a subset V' of V, the *restriction* of G to V' is the subpregraph of G over V' comprised of all edges in G whose endpoints belong to V'. This is often called an “induced subgraph” in the literature. A *subgraph* G' of a graph G is a subpregraph of G which is a graph. Note that the restriction of a graph to a subset of its vertices may or may not be a graph.

Paths. We fix a C-graph G. A path π in G is a finite or infinite sequence of edges whose endpoints match, formally

$$\pi = (v_0 \xrightarrow{c_0} v_1)(v_1 \xrightarrow{c_1} v_2)(v_2 \xrightarrow{c_2} v_3)\ldots.$$

It is very convenient to use the notation

$$\pi : v_0 \xrightarrow{c_0} v_1 \xrightarrow{c_1} v_2 \xrightarrow{c_2} \dots$$

for paths. We say that π *starts* in v_0 or that it is a path from v_0. By convention, the empty path ε starts in every vertex.

We say that vertices $v_0, v_1, v_2, \dots$ as well as edges $v_0 \xrightarrow{c_0} v_1, v_1 \xrightarrow{c_1} v_2 \dots$ *appear in* π or *are visited by* π. The *colouration* of a path π is the finite or infinite sequence of colours of edges of π, denoted $\mathrm{col}(\pi) = c_0 c_1 c_2 \dots$ for which we usually use the symbol $w \in C^{\leqslant \omega}$. We also say in this case that w is a colouration from v_0 in G.

A non-empty finite path of length $i > 0$ is of the form $\pi : v_0 \xrightarrow{c_0} v_1 \xrightarrow{c_1} \dots \xrightarrow{c_{i-1}} v_i$ and we say in this case that π is a path *from* v_0 *to* v_i, and that v_i is the *last vertex* of π. We use the notation

$$\pi : v \overset{w}{\rightsquigarrow} v'$$

to say that π is a finite path from v to v' with (finite) colouration w in G; we sometimes omit w if it is irrelevant. We also say $\pi : v \overset{w}{\rightsquigarrow} V'$ where $V' \subseteq V$ to refer to a finite path starting in v with colouration w and whose last vertex belongs to V'. A *cycle* is a non-empty path of the form $v \rightsquigarrow v$. We say that a pregraph is *acyclic* if it has no cycle.

We use the notation

$$\pi : v \overset{w}{\rightsquigarrow}$$

for an infinite path from v with colouration $w \in C^\omega$. We stress the fact that such diagrams with no last vertex always refer to infinite paths; a finite path with an unspecified last vertex would be denoted by $v \rightsquigarrow V$. When G is not clear from context, we add "in G", for instance we may write

$$\pi : v \xrightarrow{c} v' \overset{w'}{\rightsquigarrow} v'' \overset{w''}{\rightsquigarrow} \text{ in } G.$$

Finite prefixes of infinite paths define finite paths in general. Since graphs have no sinks any finite path can be extended into an infinite path therefore the converse holds: any finite path is the prefix of an infinite path. We use the notation $\Pi_{v_0} \subseteq E^*$ for the set of finite paths from v_0.

We say that a path is *simple* if no vertex is visited twice. Note that a simple path in a finite graph of size n is finite and has length $\leqslant n - 1$ (a path of length one $v_0 \xrightarrow{c} v_1$ visits two vertices). An infinite path π is a *simple lasso* if it is of the form $\pi = \pi_0(\pi_1)^\omega$ where $\pi_{<|\pi_0 \pi_1|-1}$ is simple.

Graph-morphisms. Given two C-pregraph G and G' respectively over vertices V and V', a *morphism* ϕ from G to G' is a map $\phi : V \to V'$ such that

$$v \xrightarrow{c} v' \text{ in } G \qquad \Longrightarrow \qquad \phi(v) \xrightarrow{c} \phi(v') \text{ in } G'.$$

If there exists such a morphism we say that G *maps into* G' or that G' *embeds* G. We will always be concerned with morphisms of pregraphs which are graphs, which is why we call ϕ a *graph-morphism*, even though strictly speaking it is rather a notion of pregraphs.

As an example, if G is a subgraph of G' then the inclusion $V \to V'$ defines a graph-morphism. In general, a morphism need not be injective (see Figure 6). Two graphs G, G' over V and V' are *isomorphic* if there is a bijection $\phi : V \to V'$ such that both ϕ and ϕ^{-1} define a graph-morphism. Stated differently, edges in G and G' are the same, up to renaming the vertices.

Note that if ϕ defines a morphism from G to G' then

$$v_0 \xrightarrow{c_1} v_1 \xrightarrow{c_1} \dots \text{ in } G \qquad \Longrightarrow \qquad \phi(v_0) \xrightarrow{c_1} \phi(v_1) \xrightarrow{c_2} \dots \text{ in } G',$$

therefore any colouration from v in G is a colouration from $\phi(v)$ in G'.

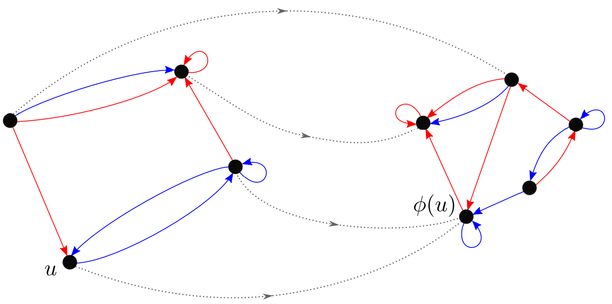


Figure 6: Two {red, blue}-graphs and a graph-morphism. Note that it is not colouration-preserving: $\phi(u)$ has red$^\omega$ as a colouration but u does not.

We say that ϕ is *colouration-preserving* if the converse holds: for all $v \in V$ and any colouration w from $\phi(v)$ in G', w is a colouration from v in G. This is a strong assumption on ϕ, which will later be relaxed.

Unordered trees, paths-graphs. A *(rooted) unordered tree* is a pregraph with a designated vertex v_0 called the *root*, and such that for every vertex v there is a unique path from v_0 to v. A sink in an unordered tree is usually called a *leaf*. Note that unordered tree are acyclic. The following theorem has numerous applications in logics and recursion theory.

Theorem 2 (König's tree lemma [Kön27])

An infinite unordered tree with finite degree has an infinite path.

Given an arbitrary graph G and a vertex v_0 we define the *paths-tree* $G_{v_0,\text{unfold}}$ of G from v_0 as the graph over the set Π_{v_0} of finite paths from v_0 in G and comprised of all edges of the form

$$(\pi : v_0 \rightsquigarrow v) \xrightarrow{c} (\pi' = \pi(v \xrightarrow{c} v')),$$

where $v \xrightarrow{c} v'$ is an edge in G. Paths-trees are trees: $G_{v_0,\text{unfold}}$ is rooted at the empty path ε, and for each vertex $(\pi : v_0 \xrightarrow{c_0} v_1 \xrightarrow{c_1} v_2 \xrightarrow{c_2} \ldots \xrightarrow{c_{i-1}} v_i)$ there is a unique path, namely

$$\varepsilon \xrightarrow{c_0} (v_0 \xrightarrow{c_0} v_1) \xrightarrow{c_1} (v_0 \xrightarrow{c_0} v_1 \xrightarrow{c_1} v_2) \xrightarrow{c_2} \ldots \xrightarrow{c_{i-1}} \pi$$

in $G_{v_0,\text{unfold}}$ from ε to π. Note that paths-trees are infinite and have no leaf since graphs have no sinks.

The map $\Pi_{v_0} \to V$ which assigns v_0 to the empty path and their last vertex to non-empty path defines a graph-morphism from $G_{v_0,\text{unfold}}$ to G which is colouration-preserving.

Graph classes. We consider collections of C-graphs, where C is fixed, which are always closed under taking subgraphs. Formally, we say that $\mathcal{C}$ is a *graph class* if for all $G \in \mathcal{C}$ and all subgraphs G' of G it holds that $G' \in \mathcal{C}$. Most often we will consider classes defined by cardinality bounds over their sets of vertices and their degree, for instance countable graphs of finite degree.

Given a class of graphs $\mathcal{C}$, we let $\mathcal{C}^{\text{paths}}$ denote the class of subgraphs of paths-trees of graphs in $\mathcal{C}$. For instance, if $\mathcal{C}$ is the class of all graphs of finite degree, then $\mathcal{C}^{\text{paths}}$ is the class of all trees of finite degree with no leaf, and therefore in this case $\mathcal{C}^{\text{paths}} \subseteq \mathcal{C}$. If however $\mathcal{C}$ is the class of all graphs of size at most 12 then $\mathcal{C}^{\text{paths}}$ is not contained in $\mathcal{C}$ since it contains (only) infinite graphs.

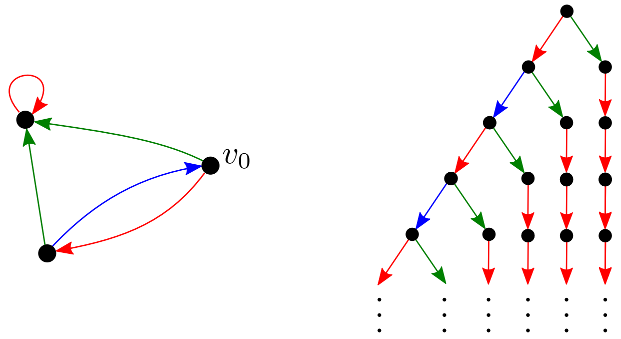


Figure 7: On the left, a graph G with a designated vertex v_0. On the right, the paths-tree $G_{v_0,\text{unfold}}$.

2 Infinite duration games on graphs

2.1 Games

We now define our main object of study, which are perfect-information, zero-sum, infinite duration, non-stochastic, graph-based, two-player, quantitative, edge-coloured games that we will simply call *games* for short.

Arenas. A C*-arena* is a C-graph together with a bipartition of its vertices. We will always use the notations V for its set of vertices. We name the players Eve (for existential $\exists$, and which will always be the minimiser) and Adam (for universal $\forall$, which will always be the maximiser) and use $V_{\text{Eve}} \sqcup V_{\text{Adam}} = V$ to denote the bipartition of the vertices.

We *always* take the point of view of Eve, and see Adam as the opponent. For instance, winning means winning for Eve. We often identify graphs with Adam-controlled arenas (formally, $V = V_{\text{Adam}}$) therefore we use the notation G both for graphs and arenas. Given a class of graphs $\mathcal{C}$, we let $\mathcal{C}^{\text{ar}}$ denote the class of all arenas whose graphs belong to $\mathcal{C}$.

We sometimes refer to vertices in V_{Eve} as Eve-vertices, and likewise for Adam. Intuitively, a game is played in a succession of moves, where a move consists in pushing a token along an edge of the arena. The choice of the edge to follow is given to the player who controls the current vertex. This interaction produces an infinite path π.

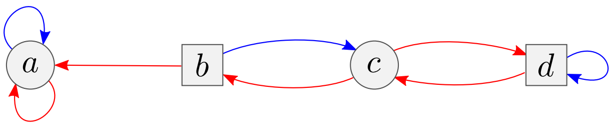


Figure 8: A $\{\text{red}, \text{blue}\}$-arena. The circle vertices a and c belong to Eve, and the two other to Adam. An example of a game played on this arena is the following: "Eve wins if eventually, the two colours alternate". If the game starts from a, b or c, Eve can ensure a win, using the following strategy: from a, Eve alternates between the two colours, and from c, Eve plays to the left. However, Adam wins if the game starts in d, by always playing the loop. We will say that $\{a, b, c\}$ is Eve's winning region. How to devise an algorithm which, given such a $\{\text{red}, \text{blue}\}$-arena, determines the winning region?

Studying a game amounts to asking whether some player can enforce a given property of the produced path, which will be defined according to its colouration. Before discussing exactly what *properties* we are interested in we formally introduce what it means to *enforce*, which requires the fundamental notion of strategies.

Strategies. Intuitively, a strategy for Eve specifies, for each possible way of reaching an Eve-vertex, what edge should be followed next.

A *prestrategy* σ for Eve (or Eve-prestrategy) from v_0 is a partial map which assigns to some paths $v_0 \rightsquigarrow v$ with $v \in V_{\text{Eve}}$ an edge $v \xrightarrow{c} v'$ in G. In words, σ picks an additional edge for some paths from v_0 which end in Eve-controlled vertices.

A finite or infinite path

$$\pi : v_0 \xrightarrow{c_0} v_1 \xrightarrow{c_1} \dots$$

is *consistent* with an Eve-prestrategy σ from v_0 if for all $i < |\pi|$ such that $v_i \in V_{\text{Eve}}$, σ is defined over $\pi_{<i}$ and

$$\sigma(\pi_{<i}) = (v_i \xrightarrow{c_i} v_{i+1}).$$

We use the notation $\pi : v_0 \overset{w}{\rightsquigarrow}_\sigma$ to say that a π is an infinite path consistent with σ with colouration w, and $\pi : v_0 \overset{w}{\rightsquigarrow}_\sigma v$ for such a finite path.

An Eve-prestrategy σ is a *strategy* if it is defined on all finite paths $\pi : v_0 \rightsquigarrow_\sigma V_{\text{Eve}}$ which are consistent with it. We refer to Figure 8 for an illustration of these important notions.

Adam-prestrategies, their consistent paths, and Adam-strategies are defined symmetrically. We generally use σ and τ respectively for strategies of Eve and of Adam. Given a pair σ, τ of strategies for each player and a starting vertex v_0, there is a unique infinite path $\pi : v_0 \xrightarrow{c_0} v_1 \xrightarrow{c_1} \dots$ from v_0 which is consistent with both strategies, which is given by

$$(v_i \xrightarrow{c_i} v_{i+1}) = \begin{cases} \sigma(\pi_{<i}) \text{ if } v_i \in V_{\text{Eve}} \\ \tau(\pi_{<i}) \text{ if } v_i \in V_{\text{Adam}} \end{cases}$$

for all $i \in \omega$. We use $\pi_{\sigma,\tau}$ to denote this path.

We let Σ_v and T_v denote the sets of strategies from v respectively for Eve and Adam. We also use $\Sigma = \bigcup_v \Sigma_v$ and $\mathrm{T} = \bigcup_v \mathrm{T}_v$.

The set of consistent paths with an Eve-strategy σ actually coincides with the set of paths realised by counter strategies τ, and vice versa.

Lemma 1 (Consistent paths and counter-strategies)

Let σ be an Eve-strategy from v_0 and let π be an infinite path starting in v_0. Then π is consistent with σ if and only if there exists an Adam-strategy τ from v_0 such that $\pi = \pi_{\sigma,\tau}$.

This offers two slightly different points of view, both of which are helpful for intuition.

Proof. The converse implication holds trivially since $\pi_{\sigma,\tau}$ is consistent with σ by definition.

Let $\pi : v_0 \xrightarrow{c_0} v_1 \xrightarrow{c_1} \dots$ be a path consistent with σ in G. Let τ be any Adam-prestrategy defined over all paths $v_0 \rightsquigarrow V_{\text{Adam}}$ (in particular, it is a strategy) and satisfying $\tau(\pi_{<i}) = (v_i \xrightarrow{c_i} v_{i+1})$ for all $i \in \omega$ such that $v_i \in V_{\text{Adam}}$. Then we have $\pi_{\sigma,\tau} = \pi$. □

Given an Eve prestrategy σ from v_0 we let Π_σ denote the set of paths from v_0 consistent with σ, and define its pregraph $G_{\sigma,\text{unfold}}$ as the restriction of the paths-graph $G_{v_0,\text{unfold}}$ of G to Π_σ. Paths in $G_{\sigma,\text{unfold}}$ from ε coincide with paths in G from v_0 which are consistent with σ (see Figure 9).

A vertex $\pi : v_0 \rightsquigarrow_\sigma V_{\text{Eve}}$ of $G_{\sigma,\text{unfold}}$ has an outgoing edge if and only if σ is defined over π, and a vertex $\pi : v_0 \rightsquigarrow_\sigma v \in V_{\text{Adam}}$ always has outgoing edges which correspond to all those of v

in G. Therefore, $G_{\sigma,\text{unfold}}$ is a graph if and only if σ is a strategy. The graph of a strategy σ contains exactly the same data as σ, and therefore provides a good way of representing (visually and mentally) an arbitrary strategy.

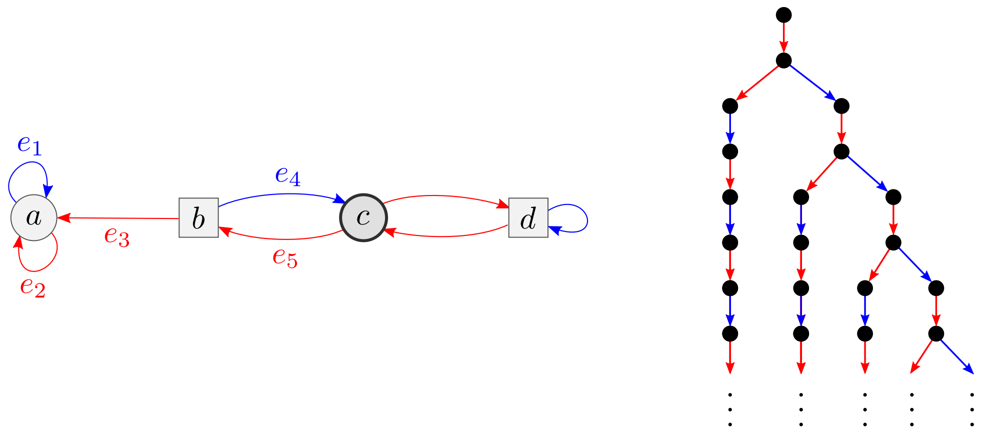


Figure 9: On the left, the {red, blue}-arena from Figure 8. We start from c, and consider the strategy described above, formally given by $\sigma(\pi) = e_5$ if $\pi = \varepsilon$ or ends in c, and otherwise $\sigma(\pi) = e_1$ if π has even length, and $\sigma(\pi) = e_2$ otherwise. Note that no path consistent with σ visits d; infinite paths consistent with σ are exactly of the form $(e_5e_4)^n e_3(e_1e_2)^\omega$ for $n \in \omega$ together with $(e_4e_5)^\omega$. On the right, the paths-graph $G_{\sigma,\text{unfold}}$, which allows to visualise these paths.

Although we will often ultimately be concerned with qualitative properties of paths, it is instrumental in our work to consider more general quantitative evaluations of paths.

Valuations and games. A *valuation over C*, assigns a *value* to any path colouration, formally

$$\text{val} : C^\omega \to X,$$

where X is a complete lattice, which we call the *set of values*. For short, we say that val is a C-valuation with values in X. A *game* G is an arena together with a valuation val.

Eve always seeks to minimise val while Adam seeks to maximise it over the produced path. Reversing the order over X produces a valuation val′ which we call the *dual* of val. This operation correspond to reversing the roles of both players.

We now formalise what it means for the players to optimize a valuation in a given game. We first extend valuations from infinite words to paths by considering their colourations, formally we let $\text{val}(\pi) = \text{val}(\text{col}(\pi))$.

Given an Eve-strategy σ from v we define the *value achieved by σ* by

$$\text{val}(\sigma) = \sup_{\tau \in \mathrm{T}_v} \text{val}(\pi_{\sigma,\tau}) = \sup_{v \overset{w}{\rightsquigarrow}_\sigma} \text{val}(w)$$

where the equality holds thanks to Lemma 1.

Optimizing over Eve-strategies allows to define the *Eve-value* of v, formally

$$\text{val}^*(v) = \inf_{\sigma \in \Sigma_v} \text{val}(\sigma).$$

This should be regarded as the optimal value of a path which Eve can guarantee, in the scenario where she should announce her strategy in advance.

Symmetrically, and with a slight abuse of notations, an Adam-strategy τ achieves at a vertex v the value

$$\mathrm{val}(\tau) = \inf_{\sigma \in \Sigma_v} \mathrm{val}(\pi_{\sigma,\tau}) = \inf_{v \overset{w}{\rightsquigarrow}_\tau} \mathrm{val}(w)$$

and we define *Adam-values* of vertices by optimizing over Adam-strategies:

$$\mathrm{val}_*(v) = \sup_{\tau \in \mathrm{T}_v} \mathrm{val}(\tau).$$

Note that the terminology refers to which player is the first to announce the strategy. We sometimes add G as a subscript when the arena is ambiguous. We say that an Eve-strategy or an Adam-strategy from v is (val-) *optimal* if its value coincides with the Eve-value or respectively with the Adam-value of v. In general, games need not have optimal strategies.

Qualitative games. When $X = \{\bot, \top\}$ is the ordered pair, we say that val is *qualitative*, and by extension that G is a qualitative game. Qualitative valuations are identified via $W = \mathrm{val}^{-1}(\bot)$ to subsets of C^ω, and in this context, we call $W \subseteq C^\omega$ the *winning condition*, or the *objective* (for Eve).

It is more convenient to work with objectives $W \subseteq C^\omega$ than qualitative valuations $\mathrm{val} : C^\omega \to \{\bot, \top\}$ therefore we generally take this point of view. Dualising a qualitative valuation corresponds to complementing the objective W.

Intuitively only two outcomes may arise, which we interpret as winning or losing; Eve wins if and only if she can guarantee to produce a path evaluated to $\bot$. We say that an infinite path π *satisfies* W if its colouration belongs to W, that a vertex v *satisfies* W if all paths from v satisfy W, and finally that a graph *satisfies* W if all its vertices satisfy W.

We say that an Eve strategy σ is *winning* if it has value $\bot$, or equivalently all paths consistent with σ satisfy W. Note that qualitative valuations have optimal strategies in general: a vertex v has value $\bot$, in which case we say that it is *winning*, if and only if there is a winning strategy from v.

2.2 Determinacy and positionality

The following result formalises the intuition that announcing one's strategy in advances gives a disadvantage in general: Eve achieves a better (smaller) value when Adam is to announce his strategy.

Lemma 2 (Comparing val^* and val_*)

We have for all $v \in V$,

$$\mathrm{val}_*(v) \leqslant \mathrm{val}^*(v).$$

Proof. Let σ_0 and τ_0 be strategies respectively for Eve and Adam from $v \in V$. We have

$$\mathrm{val}(\tau_0) = \inf_{\sigma \in \Sigma_v} \mathrm{val}(\pi_{\sigma,\tau_0}) \leqslant \mathrm{val}(\pi_{\sigma_0,\tau_0}) \leqslant \sup_{\tau \in \mathrm{T}_v} \mathrm{val}(\pi_{\sigma_0,\tau}) = \mathrm{val}(\sigma_0),$$

which concludes with the announced result by taking a supremum on the left and an infimum on the right. □

Determinacy. We say that a game is *determined* if the converse inequality holds in Lemma 2, in which case

$$\mathrm{val}_*(v) = \mathrm{val}^*(v).$$

In games which we know to be determined we refer to $\mathrm{val}_*(v) = \mathrm{val}^*(v)$ as the (optimal) *value* of v which we denote by $\mathrm{val}(v)$.

Roughly a game is determined if it does not matter which player announces their strategy first. An alternative intuition is that a game is determined if both players can *agree* on the outcome before even playing. An intuitive example is tic-tact-toe: if both players play optimally, the game can only end in a draw; therefore, good players can agree in advance that the game is a draw. As we will see below,in our setting, any *reasonable* game is determined.

We say that a valuation val is *determined* if all games with valuation val are determined. For studying determinacy, quantitative games can essentially be reduced to qualitative games. Given a valuation $\mathrm{val} : C^\omega \to X$ and $x \in X$, the *x-cut* of val is the qualitative valuation associated to $\mathrm{val}^{-1}\{X^{\leqslant x}\}$.

Lemma 3 (Reduction to qualitative case)

A valuation is determined if and only if all of its x-cuts are.

Proof. We have the following chain of implications, where G is quantified over C-arenas, v over vertices of G, and x over X.

$$\begin{array}{lll} \text{val is determined} & \iff & \forall G, \forall v, \mathrm{val}^*(v) \leqslant \mathrm{val}_*(v) \\ & \iff & \forall G, \forall v, \forall x, (\mathrm{val}_*(v) \leqslant x \implies \mathrm{val}^*(v) \leqslant x) \\ & \iff & \forall x, \forall G, \forall v, (\mathrm{val}_*(v) \leqslant x \implies \mathrm{val}^*(v) \leqslant x) \\ & \iff & \forall x, \text{ the } x\text{-cut is determined.} \end{array}$$

□

Determinacy of (qualitative) winning conditions W is a fundamental set-theoretic question, and thus has received a lot of attention since its introduction by [GS53], who proved determinacy for open and closed W. This result was later progressively extended to higher levels of the Borel hierarchy, until the seminal result of Martin [Mar75]. We do not formally introduce the Borel hierarchy but still state the theorem.

Theorem 3 (Martin's theorem)

Borel winning conditions are determined.

The theorem is essentially tight: no larger[23] topological class of conditions is determined. This result goes well beyond our needs as all examples we will consider lie within the third level of the Borel hierarchy.

Determinacy of a qualitative game rephrases as the fact that from each vertex exactly one of the players is winning. Stated differently a determined qualitative objective induces a partition corresponding to winning and losing vertices. We call this bipartition the *winning regions* respectively of Eve and Adam.

Solving a game. *Solving* a finite determined qualitative game means determining the winning regions. Solving a determined quantitative game either means determining the values of the vertices, or comparing with some value x, which amounts to solving the x-cut. By default, we take the first point of view.

[23] This can be made precise using notions that are beyond our scope. Determinacy of Gale-Stewart games plays an important role in set-theory and metamathematics.

Different extensions to infinite (but finitely presented) arenas have been considered, most prominently pushdown arenas (see the seminal work of [Wal96] and Serre's PhD thesis [Ser04], in French).

Positional determinacy. As we have seen above, arbitrary strategies are quite complicated objects (roughly, infinite trees) and in particular are often hard to describe. In many cases, such as chess, players need only to know the current configuration to make a decision; the path which led to the configuration, sometimes called *history*, is irrelevant. This allows to considerably simplify the space of strategies which are then simply (partial) maps assigning to, say, Eve-configurations an outgoing edge.

Formally, we say that a partial map $\sigma : V_{\text{Eve}} \to E$ satisfying $\sigma(v) \in \text{Out}(v)$ where it is defined is a *uniform positional prestrategy*. Given such a σ together with a starting vertex $v \in V$, we use σ_v to denote the corresponding prestrategy, formally given by

$$\sigma_v(\varepsilon) = \sigma(v) \quad \text{and} \quad \sigma_v(\pi : v \rightsquigarrow v') = \sigma(v')$$

when σ is defined in the right-hand side. We say that such a prestrategy σ_v is a *positional prestrategy*. Positional prestrategies correspond to those prestrategies which depend only on the last vertex of the path.

Note that consistency of a path with a positional prestrategy σ_v dos not depend on the starting vertex v, hence we simply say that a path is consistent with σ and denote such paths using $\rightsquigarrow_\sigma$.

A positional prestrategy σ_v induces a subpregraph G_{σ_v} of G comprised of all vertices and edges that appear in paths from v consistent with σ. We raise the reader's attention on the important distinction between the graph of a positional prestrategy (over a subset of V) and the graph $G_{\sigma',\text{unfold}}$ of an arbitrary prestrategy σ'. Nevertheless, it is again simple to verify that σ_v is a strategy if and only if G_{σ_v} is a graph.

A *uniform positional strategy* $\sigma : V_{\text{Eve}} \to E$ is a uniform positional prestrategy which is defined over all V_{Eve} (equivalently, it is a uniform positional prestrategy all of whose induced prestrategies are strategies). Given such a strategy σ, we let G_σ denote the union of the G_{σ_v}'s, which is the graph over V comprised of all edges outgoing from Adam-vertices in G and of edges in $\sigma(V_{\text{Eve}})$. Note that paths in G_σ coincide with paths consistent with σ in G.

We say that a game is *positionally determined* from v_0 if there exists a positional strategy σ_{v_0} *for Eve* from v_0 which is optimal. This corresponds with the intuition described above: Eve is able to play optimally with only the knowledge of the current configuration (or vertex), independently of the rest of the history (or path from v_0). We will study this concept in quite some depth, and many examples will arise throughout the manuscript.

We insist on the fact that the above notion is asymmetric; some authors prefer to say "positionally determined for Eve" or "half-positionally determined". We say that a game is *co-positionally determined* from v or that it is *positionally determined for Adam* from v if its dual is positionally determined from v.

We say that a game is *uniformly positionally determined* if there exists a uniform positional strategy σ for Eve which is optimal from all vertices: for all $v \in V$ it holds that $\text{val}(\sigma_v) = \text{val}(v)$. We say that a valuation val is uniformly positionally determined over a class of arenas if this is the case for all arenas of the class.

By a common abuse and only in informal discussions, we often omit the phrase "uniformly" and simply say that a given valuation is positionally determined (over a given class of arenas). This is largely justified by usage. We also use the phrase "positionality" for short to refer informally to positional determinacy. Some authors use "memoryless" strategies and valuations, which is essentially a synonym.

The question of positionality of the valuation is particularly relevant when considering the algorithmic resolution of a given finite game. Indeed a positional strategy σ can be represented over polynomially many bits and computing its values amounts to studying a graph-property over its graph G_σ, which can usualy be done in polynomial time. Therefore the problem of solving a game when the valuation is positionally determined is in NP (unless the valuation is already intractable over graphs) and likewise solving games with co-positionally determined valuations is in coNP.

Different relaxations of positionality have often been considered, among which the most natural is finite-memory determinacy (see for instance [Tho96]) which also guarantees good algorithmic properties and can be declined in several ways. Intuitively, a valuation is determined with finite memory $m \in \omega$ over a given arena if there is a machine with m states which implements an optimal strategy. Since our focus will only be on positional strategies (which correspond to the degenerate $m = 1$ case), we abstain from giving a formal definition. Note also that solving finitely presented games over infinite arenas (usually) requires finite presentations of optimal strategies, which is not guaranteed by positional (or even finite-memory) determinacy.

Infinite duration games on graphs are also sometimes defined with colours on the vertices. Although this might make a difference in some precise complexity statements the two models are often interreducible, and algorithms can often be directly transcribed from one to the other. For properties such as positional (or finite memory) determinacy however there are a few cases where the two models differ, for instance (min-) parity games with priorities in ω are bi-positionally determined only in the vertex-coloured setting, as was established by Grädel and Walukiewicz [GW06]. We will however always work in the edge-coloured setting, which is a bit more general (which is not always desirable): vertex-coloured arenas correspond to edge-coloured arenas where for all vertices, outgoing edges have the same colour.

Note that if σ_v and τ_v are positional strategies from v in a finite arena then π_{σ_v,τ_v} is a simple lasso. Therefore optimal values in finite arenas which are positionally determined for both players are reached over simple lassos.

Prefix-invariance properties. We say that a valuation val is *prefix-increasing* if for all $c \in C$ and $u \in C^\omega$ we have

$$\text{val}(cu) \geqslant \text{val}(u).$$

Stated otherwise adding a prefix increases the valuation. We say that it is *prefix-decreasing* if the other inequality holds (adding a prefix decreases the valuation), and that it is *prefix-independent* if there is an equality. Note that a qualitative valuation given by the objective $W \subseteq C^\omega$ is prefix-increasing if and only if for all colours c we have $W \supseteq cW$; it is prefix-decreasing if the converse inclusion holds, and prefix-invariant if there is an equality.

The following result if folklore; we will not use it but state it for completeness.

Lemma 4 (Uniformity of positionality for prefix-increasing objectives)

Let $W \subseteq C^\omega$ be a prefix-increasing objective. If W is positionally determined over a finite arena G then it is also uniformly positionally determined over G.

With some more care, the proof can be extended to infinite arenas.

Proof. Let $v_0 \in V$ and let σ_{v_0} be an optimal positional strategy from v_0, and consider any infinite path π in G_{σ,v_0}. Then there is a path $\pi'\pi$ from v_0 consistent with σ, thus $\text{col}(\pi'\pi) \in W$ hence by prefix-increasingness, $\text{col}(\pi) \in W$. Therefore, σ defines a winning strategy over all of G_{σ,v_0}, which contains at least one vertex, v_0. We remove G_{σ,v_0} from the arena, and conclude by induction. □

3 Some classes of games

We now define a number of different valuations which will be studied throughout the manuscript. Almost all games discussed in this manuscript fall into two important categories: ω-regular games and geometrical games. All games introduced and discussed below are determined, and most are positionally determined for Eve. Determinacy, at least for arenas of countable degree, always follows from Martin's theorem (Theorem 3) and will therefore not be discussed. Proofs of (uniform) positional determinacy, and of other results discussed below are postponed to Chapter 2.

3.1 Some ω-regular objectives

We now define a very important class of qualitative games, namely ω-regular games, which are obtained from valuations defined by ω-regular subsets of C^ω (or *languages*). These are a very robust and well-studied class of languages, which admit several equivalent definitions. We refer to [PP04] for an excellent introduction to ω-regular languages, we will only discuss a few basic ones therefore we omit a general definition.

Safety and reachability games. Safety and reachability games are in some sense the simplest non-trivial example of infinite duration games on graphs. We define the safety objective over $\{\text{safe}, \text{bad}\}$ by

$$\text{Safety} = \{\text{safe}^\omega\} \subseteq \{\text{safe}, \text{bad}\}^\omega.$$

In a safety game Eve simply aims to avoid seeing the letter bad.

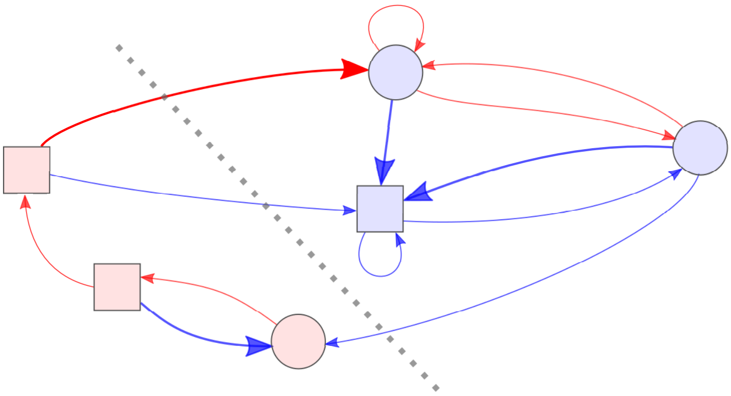


Figure 10: A finite safety game. Safe edges are depicted in blue whereas bad edges are red. The partition into winning regions is represented by the dotted line: vertices on the left are winning for Adam whereas those on the right are winning for Eve. Uniform positional winning strategies for both players are given by the bold edges.

We now define the reachability objective over $\{\text{good}, \text{wait}\}$, by

$$\text{Reachability} = \{w \in \{\text{good}, \text{wait}\}^\omega \mid |w|_{\text{good}} \geqslant 1\} = \{\text{good}, \text{wait}\}^\omega \setminus \{\text{wait}^\omega\}.$$

When playing a reachability game Eve seeks to see the letter good at least once.

Up to renaming the letters these objectives are complements of one another. Stated differently the two objectives are dual: when Eve plays a safety game, Adam plays a reachability game, and vice-versa.

Safety and reachability games are positionally determined. Finite safety and reachability games can be solved in linear time $O(m)$. The safety objective is prefix-increasing whereas the reachability objective is prefix-decreasing.

Co-Büchi and Büchi games. *Co-Büchi* games can be seen as a more resilient variant of safety games where Eve is allowed to see bad but only finitely many times. Formally they are given by the objective

$$\text{Co-Büchi} = \{w \in \{\text{safe}, \text{bad}\}^\omega \mid |w|_{\text{bad}} < \omega\}.$$

On the $\{\text{safe}, \text{bad}\}$-arena of Figure 10, the winning region for Eve in the Büchi game moreover includes the leftmost vertex, from which Eve can ensure that at most 1 occurrence of bad is seen. However the two vertices in the bottom are winning for Adam: infinitely many occurrences of bad-edges can be forced.

Dually, *Büchi* games are a harder variant of reachability games: Eve wins if she can guarantee to see the colour good infinitely many times, formally

$$\text{Büchi} = \{w \in \{\text{good}, \text{wait}\}^\omega \mid |w|_{\text{good}} = \omega\}.$$

Büchi and co-Büchi games are also known to be positionally determined. Finite Büchi and co-Büchi games can be solved in quadratic time, but no strongly subquadratic algorithm is known (see for instance [CHP08]).

If we rename colours safe, bad, wait and good respectively by $0, 1, 1$ and 2, then the co-Büchi objective is language given by words with finitely many 1's (and thus infinitely many 0's), and the Büchi objective is comprised of words with infinitely many 2's.

Parity games. Parity objectives can be seen as generalisations of Büchi and co-Büchi objectives. Colours in parity games are integers, usually called *priorities*.

What matters is the parity of the largest priority which is seen infinitely often. We will always take C to be a finite interval of integers $[a, b]$ in which case for each word in C^ω there exists a priority which has infinitely many occurrences. In a parity game, Eve should enforce that the maximal priority which is seen infinitely often is even.

Formally, we let

$$\text{Parity}_{[a,b]} = \{w \in [a, b]^\omega \mid \limsup w \text{ is even}\}.$$

For convenience, the parity game from the introduction is depicted again in Figure 11 Observe that the parity objective is invariant under adding the same even integer to all priorities, therefore without loss of generality we may assume that $[a, b]$ is of the form $[0, d]$ or $[1, d]$. We usually take d to be even for convenience.

Adding the same odd number to all priorities however complements the objective, in particular duals of parity objectives are also parity objectives. This is illustrated with the Co-Büchi and Büchi objectives, which respectively correspond to $\text{Parity}_{[0,1]}$ and $\text{Parity}_{[1,2]}$.

Parity games are positionally determined over all arenas. Moreover the parity objective is tractable over graphs and therefore the problem of solving a parity game lies in NP and also by duality in coNP.

Muller games. Muller conditions are a generalisation of parity conditions, which are defined by a subset $\mathcal{S}$ of $\mathcal{P}(C)$ where C is finite. Eve should now ensure that the set of colours which are seen infinitely often belong to $\mathcal{S}$. Formally, we let $\text{Adh}(w)$ denote the set of colours which have infinitely many occurrences in w, and put

$$\text{Muller}_{\mathcal{S}} = \{w \in C^\omega \mid \text{Adh}(w) \in \mathcal{S}\}.$$

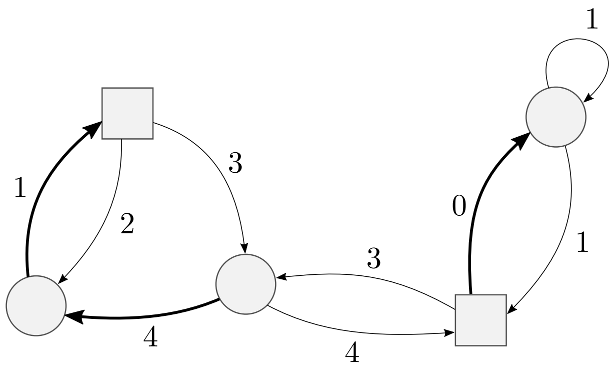


Figure 11: The parity game from the introduction.

Muller games are not positionally determined in general, but admit strategies with (fixed) finite memory even over arbitrary arenas, by the celebrated result of Gurevich and Harrington [GH82].

The problem of solving a finite Muller game is known to be PSPACE-complete from the work of Hunter and Dawar [HD05]. Note that parity conditions are special cases of Muller conditions over $C = [a, b]$ where $\mathcal{S}$ is set to be

$$\mathcal{S}_{[a,b]} = \{S \subseteq [a, b] \mid \max S \text{ is even}\}.$$

Observe that in general,

$$\text{Muller}_{{}^{c}\mathcal{S}} = {}^{c}\text{Muller}_{\mathcal{S}},$$

and that $\mathcal{S}$ is closed under unions if and only if ${}^{c}S$ is closed under intersections. Now $\mathcal{S}_{[a,b]}$ is clearly closed under unions, and so is its complement.

Rabin and Streett games. *Rabin objectives* are Muller objectives where $\mathcal{S}$ is closed under intersection, whereas their dual (by the above), *Streett objectives*, are those for which $\mathcal{S}$ is closed under union. Among all Muller conditions, Rabin objectives are exactly those which are positionally determined (and Streett conditions are thus co-positionally determined) over all arenas. Solving finite Rabin games is an NP-complete problem, while solving finite Streett games is coNP-complete as was established by Emerson and Jutla [EJ88].

Rabin conditions can also be expressed [Zie98] more conveniently as disjunctions of the form

$$\text{Rabin} = \bigcup_{i=1}^{k} \{w \mid |w|_{\text{good}_i} = \infty \text{ and } |w|_{\text{bad}_i} < \infty\},$$

over the set of colours $C = \{\text{good}_1, \text{bad}_1, \dots, \text{good}_k, \text{bad}_k\}$. In words, Eve should ensure that for some i, good_i is seen infinitely many times and bad_i only finitely many times.

Likewise, Streett conditions can be rephrased over colours $\{\text{req}_1, \text{grant}_1, \dots, \text{req}_k, \text{grant}_k\}$ by

$$\text{Streett} = \bigcap_{i=1}^{k} \{w \mid |w|_{\text{req}_i} = \infty \implies |w|_{\text{grant}_i} = \infty\}.$$

In words, Eve should ensure that for all i, if the i-th request req_i is seen infinitely often, then it must be granted infinitely often.

Note that Muller conditions, and therefore co-Büchi, Büchi, parity, Rabin and Streett objectives, are all prefix-independent. It is not the case that prefix-independent ω-regular objectives are Muller objectives; for instance the objective "eventually alternate between blue and red" is prefix-independent and ω-regular but not Muller.

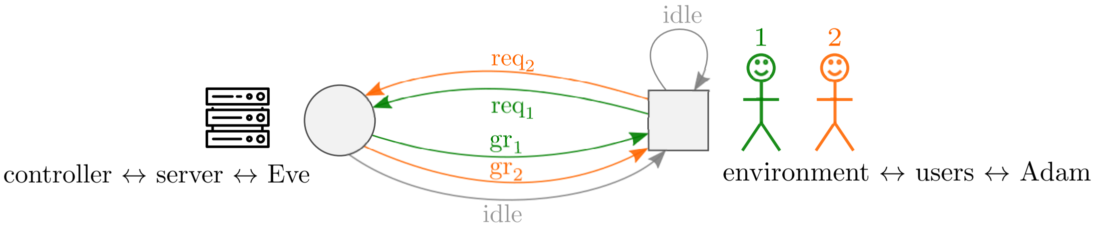


Figure 12: A Streett game modelling a simple reactive synthesis scenario. Note that Eve wins (a two-state controller can be synthesised) however there is no winning positional strategy.

3.2 Payoff games

Unlike ω-regular games which are inherently qualitative, payoff games naturally arise as arenas equipped with quantitative valuations. Here colours are real numbers, called *weights* or *payoffs*, which we denote by $t \in \mathbb{R}$. We often restrict to subsets of the reals, which we indicate on valuations as subscripts, for instance $\mathrm{val}_{\mathbb{Z}}$, and very often consider bounded integer weights, thus for convenience we will write val_N for $\mathrm{val}_{[-N,N]_{\mathbb{Z}}}$, where $N \in \omega$.

Mean-payoff games. Mean-payoff games are arenas over bounded sets of weights equipped with the valuation

$$\mathrm{MP}(t_0 t_1 \dots) = \limsup_k \frac{1}{k} \sum_{i=0}^{k-1} t_i.$$

Here, weights are seen as payoffs from Eve to Adam, and Eve seeks to minimize the average payoff in the long run. Mean-payoff games are positionally determined for both players over finite arenas (we refer to the introduction for more discussion and references). A finite mean-payoff game is discussed in Figure 15. Over infinite arenas there might be no optimal strategy for either player (see Figure 13).

Over finite games values are reached with positional strategies for both players [EM73; EM79], which produce paths that are ultimately cycling. Now the colouration w of such a path has the mean value of the cycle as its mean-payoff, and in particular the lim sup in the definition can equivalently be replaced by a lim inf.

Stated differently, thanks to positionality and only over finite games, dualising the mean-payoff valuation simply amounts to inverting the signs of the weights

$$\begin{aligned} \text{maximizing } \mathrm{MP}(w) \quad &\iff \quad \text{minimizing } -\mathrm{MP}(w) \\ &\iff \quad \text{minimizing } \mathrm{MP}(-w), \end{aligned}$$

where the second equivalent holds by the above discussion (lim sup can be replaced with lim inf for optimal paths). Therefore, just like parity games, mean-payoff games over finite games are symmetric: up to changing the signs, the two players play the same role. They are moreover tractable over graphs hence the problem of solving a mean-payoff game lies in NP ∩ coNP.

Energy games. Energy games are arenas equipped with the valuation

$$\mathrm{Energy}^+(t_0 t_1 \dots) = \sup_k \sum_{i=0}^{k-1} t_i \in [0, \infty].$$

In words, Eve aims to minimize the highest value reached by the profile. Note that although they are antagonistic (as always), the roles of the two players are asymmetric. Energy games with weights in $\mathbb{Z}$

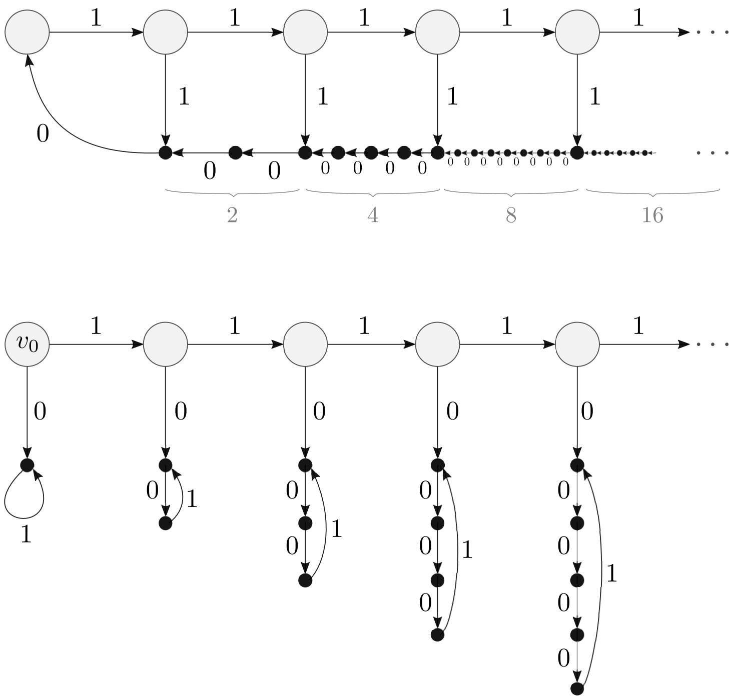


Figure 13: Two infinite mean-payoff game of degree 2. For vertices with only one outgoing edge it does not matter whether they belong to Eve or Adam; therefore we write them as graph vertices for readability (we often use this notational convention throughout the thesis). For the game in the top, all vertices have value 0, and there exists an optimal strategy ensuring it (go further and further away to the right), however positional strategies have positive values.
For the game in the bottom, the value of v_0 is 0 however there is no optimal strategy: any (even non-positional) strategy has positive value.
As mentionned above, non-existence of optimal strategies is not possible for quantitative games. In the bottom arena, Eve actually loses the threshold mean-payoff game $\mathrm{MP}^{\leqslant 0}$ even though v_0 has value 0.

are positionally determined in general and co-positionally determined over finite arenas [BFL+08]; this is illustrated in Figure 14. (Arbitrary energy games with weights in $\mathbb{Q}$ are not positionally determined, an example is given by a single Eve vertex with all $(0,1]$-loops.) Finite energy games are therefore solved in NP $\cap$ coNP [BFL+08].

We like to interpret edge weights as changes in temperature: positive weights make the temperature warmer, whereas negative weights make it colder. Eve's goal is then to keep to temperature bounded along the game. The optimal value is the least temperature upper bound Eve can guarantee. A finite energy game is discussed in Figure 15.

A more common analogy which justifies the name is that of a battery: starting from some energy level, and interpreting positive weights as battery depletion and vice-versa, Adam seeks to empty the battery while Eve wants to keep it running. The optimal value then corresponds to the smallest initial energy level from which Eve can guarantee to keep the battery above zero, and ∞ if there is none.

Threshold mean-payoff. We refer to the objective $\mathrm{MP}^{\leqslant 0}$ as the *threshold mean-payoff* objective. Note that any word with positive MP has an unbounded profile, or in the contrapositive

$$\mathrm{Energy}^{+}(w) < \infty \qquad \Longrightarrow \qquad \mathrm{MP}(w) \leqslant 0.$$

In a finite arena of size n, a simple lasso has non-positive mean-payoff if and only if it has bounded

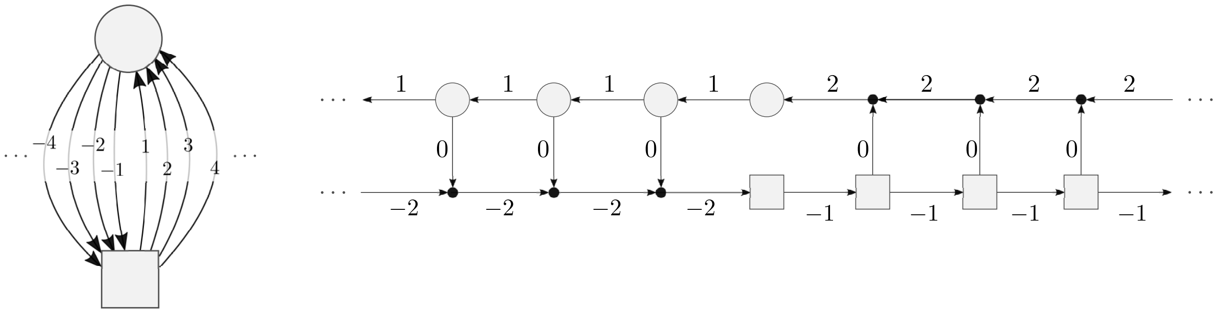


Figure 14: Two infinite arenas. The one on the left has infinite degree and infinitely many different weights whereas the one on the right has degree 2 and weights in $\{-2,-1,0,1,2\}$. Both energy games are similar: Adam has strategies to ensure value ∞ from all vertices (always do more than Eve), however all positional strategies have finite value.

energy, if and only if it has energy $\leqslant (n-1)N$.

Therefore (thanks to positionality) threshold mean-payoff games are closely related with energy games: over finite arenas, the winning region for $\mathrm{MP}^{\leqslant 0}$ coincides with the set of vertices with finite Energy^{+}.

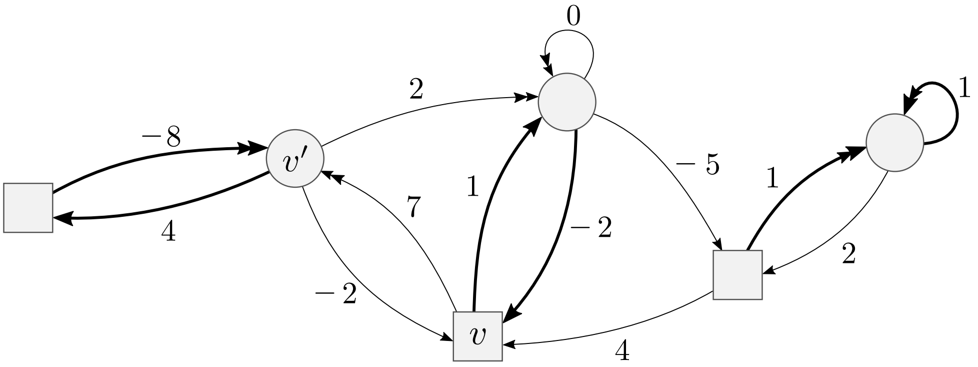


Figure 15: The example from the introduction, adapted so as to fit our working convention. We repeat and adapt the explanation for convenience and readability. Mean-payoff values from left to right are $-2,-2,-\frac{1}{2},-\frac{1}{2},1$ and 1, and mean-payoff-optimal positional strategies for both players are identified in bold. Energy values are $0,2,9,0,\infty$ and ∞, and energy-optimal strategies are given by arrows with double heads. Notice that from v, Adam can ensure that the temperature reaches at least 9.
However to do so he must take the edge towards v', which is non-optimal with respect to mean-payoffs: it gives Eve the possibility to ensure a long term average of -2 by forcing the leftmost cycle.
Notice also that using a mean-payoff-optimal strategy from v' ensures a finite upper bound, namely 4, on the temperature; it is however not optimal. Actually, mean-payoff-optimal strategies for Eve are also viable in the energy game in general, in the sense that they achieve a finite (but possibly non-optimal) energy-value.

Finite parity games of size n can be reduced to threshold mean-payoff games simply by replacing each priority p by the weight $(-n)^{p+1}$. The validity of this reduction again follows simply from positional determinacy since a simple lasso has even lim sup if and only if if has positive mean-payoff.

Discounted games and reductions. Discounted games are arenas with bounded weights equipped with the valuation

$$\mathrm{Disc}_\lambda(w) = \sum_{i=0}^{\infty} \lambda^i w_i,$$

where $0 < \lambda < 1$ is a fixed parameter. Intuitively, more importance is now given to weights which are visited sooner. Discounted games of finite degree are positionally determined for both players, as can be derived from the Shapley's seminal paper [Sha53] on stochastic games. If λ is fixed, discounted games can be solved in strongly polynomial time $O((\frac{m}{1-\lambda}\log(\frac{n}{1-\lambda}))^2)$ by strategy improvement (even in the stochastic setting) as was recently established by Hansen, Miltersen and Zwick [HMZ13], building on the results of Ye [Ye11].

If λ is taken to be close enough to one (details in Chapter 2), positivity of the discounted value of a simple lasso is equivalent to the positivity of its mean-payoff; therefore finite threshold mean-payoff games can be reduced to finite discounted games. Combined with the reduction discussed above, one may also reduce finite parity games to discounted games.

The mean-payoff valuation is prefix-invariant (and therefore so is $\mathrm{MP}^{\leqslant 0}$) but energy and discounted valuations are not.

Part I

Well-monotonic graphs and positionality

Introduction

Understanding memory requirements – and in particular positionality – of given valuations or objectives has been a deep and challenging endeavour dating back at least to the work of Shapley [Sha53] (for finite concurrent stochastic games) and then of Büchi and Landweber [BL69], Büchi [Büc77] and Gurevich and Harrington [GH82] which are more closely related to our setting. Among others, the seminal works of Shapley [Sha53], Ehrenfeucht and Mycielski [EM73], and later Emerson and Jutla [EJ91], Klarlund [Kla92], McNaughton [McN93] and Zielonka [Zie98], have given us a few important ideas and tools, which were later redigested, enhanced and extended on numerous occasions.

Very roughly speaking (more details are provided below), these early efforts culminated in Gimbert and Zielonka's [GZ05] complete characterisation of bi-positionality over finite arenas on one hand (see also Gimbert's PhD thesis [Gim07], in French), and Kopczyński's [Kop06] general results and conjectures on positionality on the other. In the recent years, increasingly expressive and diverse valuations and objectives have emerged from the fast-paced development of reactive synthesis, triggering more and more interest in these questions.

By now, bi-positionality is well understood, and the frontiers of finite-memory determinacy are becoming clearer. However, recent general approaches to finite-memory determinacy often behave badly when instantiated to the degenerated case of positionality (memory one), for different reasons which are detailed below. Therefore, and walking in the footsteps of Klarlund, Kopczyński and others, we propose a generic tool for (half) positionality, and moreover present a new characterisation result. Before discussing our approach, we briefly survey the state of the art, with a focus on integrating several recent and successful works in different related settings.

Bi-positionality. The celebrated result of Gimbert and Zielonka [GZ05] characterises valuations which are bi-positional over finite arenas (including parity objectives, mean-payoff, energy, and discounted valuations, and many more). The characterisation is most useful when stated as follows (one-to-two player lift): a valuation is bi-positional if (and only if) each player has optimal positional strategies on arenas *which they fully control*. Therefore, bi-positionality is reduced to a property of graphs. In this regard, our main result in Part I is analogous.

Bi-positionality over infinite arenas is also well understood thanks to the work of Colcombet and Niwiński [CN06], who established that any prefix-independent objective which is bi-positional over arbitrary arenas is, up to renaming the colours, a parity condition (with finitely many priorities). Two remarks are in order here. First, this result was already known[1] for Muller objectives from the work of Zielonka [Zie98], discussed below. Second, this gives a sharp contrast with the vertex-coloured case, in which Grädel and Walukiewicz [GW06] have established bi-positionality of several other prefix-independent conditions, most notably the min-parity condition with ω priorities.

[1] For finite arenas, even from McNaughton [McN93]. Also, Zielonka formally proves the result only for arenas of finite degree, but notes that the assumption is not required.

Very recently, Kozachinskiy [Koz21a] has given a thorough study of the particular case of *continuous* valuations $A^\omega \to \mathbb{R}$ over finite arenas. Intuitively, the assumption of continuity is orthogonal to prefix-independence: here, values are obtained as limits over finite prefixes. Among the valuations evoked so far, the discounted valuation is the only one that is continuous. Although the characterisation of [GZ05] applies here, Kozachinskiy provides several novel insights which we will later relate to our work on different occasions.

Finite-memory determinacy. For applications in synthesis, establishing finite-memory determinacy as well as determining minimal finite-memory requirements are fundamental since such strategies correspond to programs. Finite-memory determinacy of Muller games over finite arenas was first established by Büchi and Landweber [BL69], and the result was extended to infinite arenas by Gurevich and Harrington [GH82]. Zielonka [Zie98] was the first to investigate precise memory requirements and he introduced what Dziembowski, Jurdziński and Walukiewicz [DJW97] later[2] called the *Zielonka tree* of a given Muller condition, a data structure which they used to precisely characterise its memory requirement.

Another general characterisation of finite memory requirements was given by Colcombet, Fijalkow and Horn [CFH14] for *generalised safety* conditions over arenas of finite degree, which are those defined by excluding an arbitrary set of prefixes of colours (topologically, Π_1). This characterisation is orthogonal to the one for Muller conditions (which are prefix-independent); it provides in particular a proof of positionality for (threshold) generalisations of energy objectives, and different other results.

Le Roux, Pauly and Randour [RPR18] identified a sufficient condition ensuring that finite memory determinacy over finite arenas is preserved under boolean combinations. Although they encompass numerous cases from the literature, the obtained bounds are generally not tight, and in particular no generic result for combinations of positional objectives can be extracted from their work.

We mention also a recent general result of Bouyer, Le Roux and Thomasset [BLT21], in the much more general setting of (graph-less) *concurrent* games given by a condition $W \subseteq (A \times B)^\omega$: if W belongs to Δ_2^0 and residuals form a well-quasi order[3], then it is finite-memory determined[4]. We will also rely on well-founded orders (although ours are total), but stress that our results are incomparable: to transfer the result of [BLT21] to game on graphs, one encodes the (possibly infinite) arena in the winning condition W, and therefore strategies with reduced memory no longer have access to it. This is not an issue if G is finite (and if one complies with having memory bounds depending on n) for the case of finite memory, but positionality results cannot be transferred. Moreover, we will often care about infinite arenas.

Chromatic and arena-independent memories. Kopczyński [Kop06] proposed to consider strategies implemented by memory-structures that depend only on the colours seen so far (rather than on the path), which he called *chromatic* memory – as opposed to usual *chaotic* memory. His motivations for studying chromatic memory are the following: first, it appears that for several (non-trivial) conditions, chromatic and chaotic memory requirements actually match; second, any ω-regular condition W admits optimal strategies with finite chromatic memory, implemented by a deterministic (parity or Rabin) automaton recognising W; third, such strategies are arena independent, and one may even prove (Proposition 8.9 in [Kop06]) that in general, there are chromatic memories

[2] It appears that technical reports of Zielonka's work were accessible as early as 1994.

[3] We will abstain from giving formal definitions for these two notions. Since the inclusion relation is antisymmetric, it is a well-quasi order if and only if it is a well-partial order (the notion used by the authors).

[4] In the concurrent setting, games are often not even determined (even when Borel). This is not an issue for considering finite-memory determinacy, which means "if a winning strategy exists, then there is one with finite memory".

of minimal size which are arena-independent. Kopczyński therefore poses the following question: does it hold that chromatic (or equivalently, arena-independent) and chaotic memory requirements match in general?

This turns out not to be the case, a (non ω-regular) counterexample being given by multi energy objectives, which have finite chaotic memory strategies but require infinite chromatic memory. A very recent work of Casares [Cas21], studies this question for Muller games, for which an elegant characterisation of chromatic memory is given: it coincides with the size of the minimal transition-coloured deterministic Rabin automaton recognising it. Comparing with the characterisation of [DJW97] via Zielonka trees reveals a gap between arena-dependent and independent memory requirements already for Muller conditions.

Arena-independent (finite) memory structures have independently been investigated recently by Bouyer, Le Roux, Oualhadj, Randour and Vandenhoven [BLO+20] over finite arenas. In this context, they were able to generalise the characterisation of [GZ05] (which corresponds to memory one), to arbitrary memory structures. As a striking consequence, the one-to-two player lift of [GZ05] extends to *arena-independent* finite memory: if both players can play optimally with finite arena-independent memory respectively n_{Eve} and n_{Adam} in one-player arenas, then they can play optimally with finite arena-independent memory $n_{\text{Eve}} \cdot n_{\text{Adam}}$ in general. A counterexample is also given in [BLO+20] for one-to-two player lifts in the case of arena-dependent finite memory.

Many valuations and objectives considered for synthesis admit arena-independent finite memory. This characterisation was more recently generalised to pure arena-independent strategies in stochastic games by Bouyer, Oualhadj, Randour and Vandenhoven [BOR+21], and even to concurrent games *on graphs* by Bordais, Bouyer and Le Roux [BBR21]. Unfortunately, none of these result carry over well to (half) positionality, since they inherit from [GZ05] the requirement that *both players* rely on the same memory structure. For instance, in a Rabin game, the antagonist requires finite memory > 1 in general, and therefore the results of [BLO+20] cannot establish positionality.

Positionality. Unfortunately, there appears to be no[5] characterisation similar to Gimbert and Zielonka's for (half) positionality. In fact, there has not been much progress in the general study of positionality since Kopczynśki's work, on which we now briefly comment.

Kopczyński's main conjecture [Kop06] on positionality is the following:

"Prefix-independent positional objectives are closed under finite union."

which we will call Kopczyński's conjecture. It can be instantiated either for positionality over arbitrary arenas, or only finite arenas, leading to two incomparable variants both of which are open, even for countable unions. An elegant counterexample to a stronger statement is presented in [Kop06]: there are uncountable unions of Büchi conditions which are not positional over some countable arenas. Kopczyński introduces two classes[6] of prefix-independent objectives which are positional and closed under finite unions: *concave* objectives and *monotonic* objectives.

Concave objectives are complements of *convex* objectives, which are those closed under shuffles (not defined here). It is immediate that convex objectives are closed under arbitrary intersection, and therefore concave ones under arbitrary unions. Examples include the parity objective, and the threshold mean-payoff objective (here, the lim sup semantics is important). The main result is that concave objectives are positional in general over *finite* arenas; of course this is not true for infinite arenas, for instance because of mean-payoff objectives. This result was later extended to also encompass some non-prefix independent objectives by Bianco, Faella, Mogavero and Murano [BFM+11], but closure under union is lost.

[5] A formal statement can be found in [Kop06], page 34.

[6] Actually, there are more, but we only discuss these two here.

Monotonic objectives are those of the form $C^\omega \backslash \mathcal{L}^\omega$, where $\mathcal{L} \subseteq C^*$ is a (regular) language recognised by a linearly ordered deterministic automaton[7] whose transitions are monotonous. Monotonic objectives are prefix-independent, closed under finite unions, and shown to be positionally determined over *arbitrary* arenas [Kop06]. Our work builds on Kopczyński's suggestion to consider well-ordered automata; however to obtain a complete characterisation we crucially replace the automata-theoretic semantic of *recognisability* by the graph-theoretical *universality* which is more adapted to the fixpoint approach we pursue. We discuss in the conclusion of Part I how our notions instantiates to Kopczyński's.

Our approach. We introduce *well-monotonic* graphs, which are well-ordered graphs over which each edge relation is monotonic, and prove in a general setting that existence of *universal* well-monotonic graphs implies positionality. The idea of using adequate well-founded (or ordinal) measures to *fold* arbitrary strategies into positional ones is very natural, and far from being novel: it appears in the works of Emerson and Jutla [EJ91] (see also Walukiewicz' presentation [Wal96], and Grädel and Walukiewicz' extensions [GW06]), but also of Zielonka [Zie98] (and in a completely different way! More discussion about this in Chapter 11) for parity games, and was even formalised by Klarlund [Kla91; Kla92] in his notion of *progress measures* for Rabin games.

In this context, the universality assumption is transparent: it simply states the existence of a progress measure. For instance, the impressive proof of Klarlund and Kozen [KK91] can be somewhat artificially rephrased as a universality result for an (involved) monotonic graph with respect to the Rabin condition, which was later used by Klarlund [Kla92] to establish their positionality.

Our first contribution here is simply conceptual and consists in streamlining the argument, and in particular expliciting the measuring structure as a (well-monotonic) graph[8]. We believe that this has two advantages.

(i) Separating the strategy-folding argument from the universality argument improves conceptual clarity. In particular, we believe that known proofs are seen in a new light, and we also extend a few known results.

(ii) Perhaps more importantly, well-monotonic graphs then appear as concrete and manageable witnesses for positionality. One can imagine many different ways of combining them (we recall that ordinals can be added, multiplied or even exponentiated). Moreover, different meaningful subclasses of well-monotonic graphs leading to as many interesting classes of positional objectives (among them, Kopczyński's monotonic objectives) can be envisaged.

We supplement (ii) with our main technical novelty: any positional valuation which has a neutral letter[9] admits universal well-monotonic graphs. Stated differently, for valuations with a neutral letter, existence of universal well-monotonic graphs characterises (half) positionality. Such a characterisation result is completely novel; it holds in the qualitative setting, with no prefix-independence assumption.

Finally, as a proof of concept and inspired by Walukiewicz's presentation [Wal96] of Emerson and Jutla's proof [EJ91], we show that universality of well-monotonic graphs is preserved by finite

[7] The automaton is assumed to be finite, but Kopczyński points out (page 45 in [Kop06]) that the main results still hold whenever the state space is well-ordered and admits a maximum (stated differently, it is a non-limit ordinal).

[8] The recent line of work of Bouyer and co-authors discussed above is analogous in this regard: the finite memory structure $\mathcal{M}$ is considered *externally*, independently of the arena. However, their scope (bi-finite determinacy), techniques and results are different.

[9] This notion is defined in Chapter 3.

lexicographical products of prefix-independent objectives[10]. Thanks to our characterisation result, this implies that prefix-independent positional objectives with a neutral letter are closed under lexicographical product. (In this scenario, the parity condition can be obtained as a lexicographical product of Büchi or of co-Büchi conditions.) Our hope, which will be further discussed in the conclusion of Part I, is that similar constructions can be employed to make progress on Kopczyński's conjecture.

Organisation of Part I. In Chapter 1 we introduce monotonic graphs and universality, then present the positionality result and its proof. We include a discussion on how the framework instantiates to the important case of prefix-independent objectives.

Chapter 2 is concerned with manipulating well-monotonic graphs. We first illustrate the new notions over several examples (safety, reachability, Büchi, co-Büchi and energy), for which we give general positionality proofs and templates for proving universality. Second, we turn to discounted and mean-payoff games, which are not positional over arbitrary arenas, and discuss their positionality (which falls out of the scope of our technique; we see this in a positive light). Third, we study general variants of counter games for which well-monotonic graphs are available, establishing their positionality. Last, we provide a generic construction for lexicographical products.

Chapter 3 presents our main novelty in terms of a *structuration* result: roughly, if val is positional, then any graph can be equipped with a well-monotonic structure simply by adding edges, and – crucially – without increasing val. This implies the wanted converse statement, therefore establishing the characterisation.

We actually prove two different structuration results. The first one relies on *saturation* and is inspired by Colcombet and Fijalkow's work [CF19], but it only works for finite graphs; it will be used in subsequent Chapters, but is not helpful for the wanted characterisation. The second result requires a stronger hypothesis on the neutral letter, and relies on a natural object we called *multiple choice arenas*, which exploit the positionality hypothesis in a much more efficient fashion.

Before turning to algorithms for the remainder of the thesis, we conclude the first part by discussing our perspectives for the study of positionality.

We express our gratitude to Thomas Colcombet for several important insights, in particular the "need for non-uniformity" and "limits of saturation", which turned out to be instrumental for this first part.

[10] This requires a construction for lexicographical combinations of well-monotonic graphs, which is naturally supported by ordinal multiplication.

Positionality from well-monotonic graphs 1

This chapter lays the foundations for Part I and the rest of the thesis. More precisely, Section 1 introduces monotonic graphs and universality, our two most important concepts. In Section 2, it is shown how positionality results can be deduced from the existence of universal well-monotonic graph. Last, Section 3 discusses how these notions instantiate to the important special case of prefix-independent objectives.

1 Monotonic graphs and universality

1.1 Monotonic graphs

Definition. A C-graph $\mathcal{L}$ is *monotonic* if its vertex set L is equipped with a linear order $\geqslant$ which is well behaved with respect to the edge relation in the sense that in $\mathcal{L}$,

- if $\ell \geqslant \ell' \xrightarrow{c} \ell''$ then $\ell \xrightarrow{c} \ell''$, and
- if $\ell \xrightarrow{c} \ell' \geqslant \ell''$ then $\ell \xrightarrow{c} \ell''$.

The first item states that the order is well behaved "at the left" of edges, while the second refers to its behaviour "at the right"; we thus refer to the first property as *left composition*, and to the second one as *right composition*. See Figure 1.1 for an illustration.

Note that monotonicity is preserved by taking induced subgraphs, with the induced linear order. Observe also that in a monotonic graph $\mathcal{L}$, if $\ell \geqslant \ell'$ then ℓ has more colourations than ℓ': a path $\ell' \xrightarrow{c} \ell'' \overset{w'}{\rightsquigarrow}$ with colouration $w = cw'$ from ℓ' in $\mathcal{L}$ induces by left composition a path, namely $\ell \xrightarrow{c} \ell'' \overset{w'}{\rightsquigarrow}$, with the same colouration from ℓ. We refer to this property as *colouration-monotonicity*.

Successors, predecessors, completeness. Let $\ell \in L$ and $c \in C$, and consider

$$\Delta(\ell, c) = \{\ell' \in L \mid \ell \xrightarrow{c} \ell' \text{ in } \mathcal{L}\},$$

the set of c-successors of ℓ in $\mathcal{L}$. Right composition states exactly that $\Delta(\ell, c)$ is downward-closed.

Likewise, for any $\ell' \in L$ and $c \in C$, the set[1]

$$\mathrm{P}(\ell', c) = \{\ell \in L \mid \ell \xrightarrow{c} \ell' \text{ in } \mathcal{L}\}$$

[1] We read P as a capital rho.

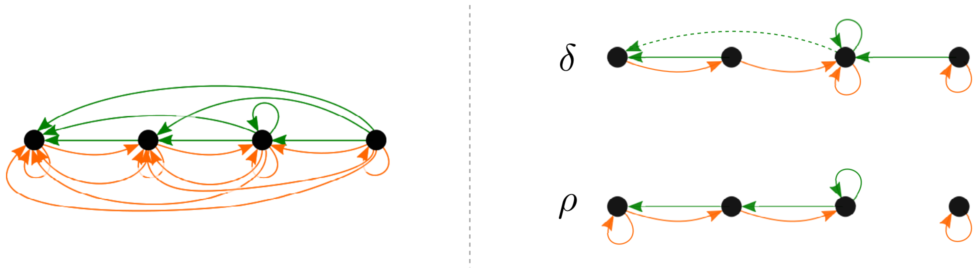


Figure 1.1: On the left, a finite monotonic graph with two colours, the order is increasing from left to right (this convention is adopted throughout the thesis). Note that it is rather dense; monotonic graph often have a quadratic number of edges. When depicting them, we generally omit many edges which follow from composition. Two examples representing the same monotonic graph while displaying fewer edges are given on the right (they correspond to max-successors and min-predecessors); for example the green dashed edge follows from the green loop by right composition.

is upward-closed, thanks to left composition.

We say that a monotonic graph $\mathcal{L}$ is *completely monotonic* if $\leqslant$ defines a complete linear order over L and moreover the maximal element $\top \in L$ is such that $\top \xrightarrow{c} \ell$ for all $\ell \in L$ and $c \in C$. Stated differently, all vertices have a predecessor for each colour, namely $\top$. In a monotonic graph with a maximal element $\top$, since $\mathrm{P}(\ell', c)$ is upward closed, having a predecessor amounts to having $\top$ as a predecessor: completeness of a monotonic graph thus corresponds to order-completeness of $\geqslant$ and automata-theoretic co-completeness of the graph (each vertex has a predecessor for each colour).

For a completely monotonic graph $\mathcal{L}$ over L, and for $\ell, \ell' \in L$ and $c \in C$, we define

$$\delta(\ell, c) = \sup \Delta(\ell, c) \in L \quad \text{ and } \quad \rho(\ell', c) = \inf \mathrm{P}(\ell', c) \in L,$$

which we respectively call the *sup-successor* and *inf-predecessor* tables of $\mathcal{L}$. Note that $\delta(\ell, c)$ needs not be a c-successor of ℓ and likewise for ρ.

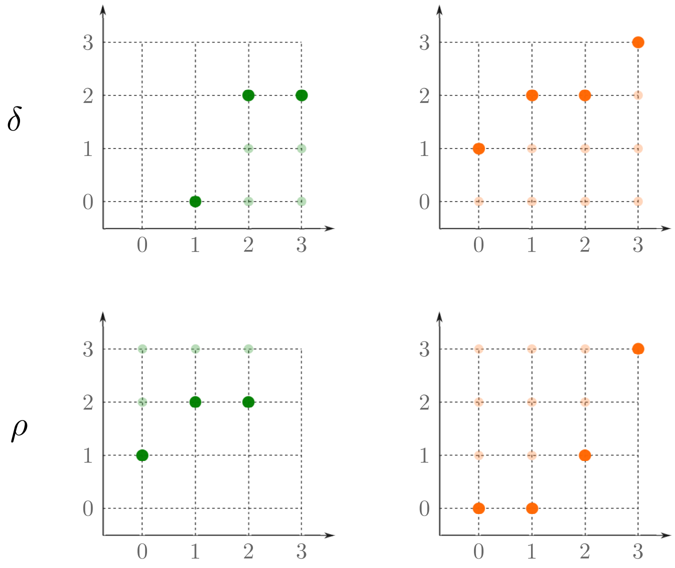


Figure 1.2: The successor and predecessor tables of the monotonic graph displayed in Figure 1.1 for each of the two colours. Sup-successors and inf-predecessors are in bold.

Given any C-graph $\mathcal{L}$ equipped with a complete linear order, we say that δ *is defined* if the $\Delta(\ell, c)$'s are downward-closed, and that ρ *is defined* if the $\mathrm{P}(\ell', c)$'s are upward-closed. For the purpose of stating the lemma below, we let δ_c and ρ_c be given by $\delta_c : \ell \mapsto \delta(\ell, c)$ and $\rho_c : \ell' \mapsto \rho(\ell', c)$ from L to L.

Lemma 1.1 (Monotonicity in complete graphs)

Let $\mathcal{L}$ be equipped with a complete linear order. Then

$$\begin{aligned} \mathcal{L} \text{ is monotonic} \quad &\iff \quad \delta \text{ is defined and the } \delta_c\text{'s are monotonic} \\ &\iff \quad \rho \text{ is defined and the } \rho_c\text{'s are monotonic.} \end{aligned}$$

The conditions on the right are often easier to verify than left and right composition and provide an alternative point of view.

Proof. We prove the first equivalence, the second one is dual. We have already seen that δ is defined if and only if $\mathcal{L}$ has right composition.

Let $c \in C$ and $\ell \geqslant \ell'$ in L. left composition states that c-successors of ℓ' are also c-successors of ℓ, or stated differently $\Delta(\ell, c) \supseteq \Delta(\ell', c)$. Thus if $\mathcal{L}$ is monotonic, we have $\sup \Delta(\ell, c) \geqslant \sup \Delta(\ell', c)$, which proves the monotonicity of δ_c. Conversely, since δ is defined, Δ's are downward-closed, hence $\sup \Delta(\ell, c) \geqslant \sup \Delta(\ell', c)$ implies $\Delta(\ell, c) \supseteq \Delta(\ell', c)$ which rephrases left composition. This proves the first equivalence, and the second holds by symmetry. □

Progress measures. We now fix a C-arena G over V and a completely monotonic graph $\mathcal{L}$ over L. A *progress measure* (over G in $\mathcal{L}$) is a map $\phi : V \to L$. Progress measures are (partially) ordered pointwise:

$$\phi \geqslant \phi' \qquad \iff \qquad \forall v, \phi(v) \geqslant \phi'(v).$$

We define the *(global) backpropagation operator (for Eve)* over progress measures by

$$\begin{aligned} \mathrm{Upd}_G^{\mathcal{L}}(\phi)(v) \quad &= \quad \begin{cases} \inf\limits_{v \xrightarrow{c} v' \text{ in } G} \inf\{\ell \in L \mid \ell \xrightarrow{c} \phi(v') \text{ in } \mathcal{L}\} \text{ if } v \in V_{\text{Eve}} \\ \sup\limits_{v \xrightarrow{c} v' \text{ in } G} \inf\{\ell \in L \mid \ell \xrightarrow{c} \phi(v') \text{ in } \mathcal{L}\} \text{ if } v \in V_{\text{Adam}} \end{cases} \\ &= \quad \begin{cases} \inf\limits_{v \xrightarrow{c} v' \text{ in } G} \rho(\phi(v'), c) \text{ if } v \in V_{\text{Eve}} \\ \sup\limits_{v \xrightarrow{c} v' \text{ in } G} \rho(\phi(v'), c) \text{ if } v \in V_{\text{Adam}}, \end{cases} \end{aligned}$$

where the equality holds by definition of ρ. We usually drop the subscript and/or the superscript when G and $\mathcal{L}$ are fixed and clear from context.

The intuition behind the definition is rooted in the idea of *simulating* paths in the arena G using paths in the monotonic graph $\mathcal{L}$: $\mathrm{Upd}(\phi)(v)$ represents the smallest position ℓ in $\mathcal{L}$ such that Eve can ensure that the next edge which is visited in the arena belongs to $\mathcal{L}$. Note that this intuition directly suggests a positional strategy (assuming the inf is met) where, from v, Eve chooses an edge minimising $\rho(\phi(v'), c)$.

Since $\mathcal{L}$ is complete, the pointwise order over L^V equips the set of progress measures with the structure of a complete lattice. By monotonicity of ρ, the above operator is monotonic. Hence by the Knaster-Tarski theorem (Theorem 1), its set of fixpoints forms a complete lattice, and its least fixpoint coincides with its least prefixpoint.

1.2 Morphisms preserving val and universality

Fix a valuation val : $C^\omega \to X$. Recall that values of vertices in a graph G are defined by seeing them (as always) as Adam-controlled arenas, formally

$$\mathrm{val}_G(v) = \sup_{v \overset{w}{\rightsquigarrow} \text{ in } G} \mathrm{val}(w).$$

Given two graphs G and G' with a graph morphism $\phi : G \to G'$, since there are more colourations from $\phi(v)$ in G' than from v in G we have in general

$$\mathrm{val}_G(v) \leqslant \mathrm{val}_{G'}(\phi(v)).$$

Preservation of val. We say that ϕ *preserves* val or that it is a val-*preserving* morphism if the converse inequality holds: for all $v \in V$, $\mathrm{val}_{G'}(\phi(v)) \leqslant \mathrm{val}_G(v)$. Note that a colouration-preserving morphism is always val-preserving; stated differently, being val-preserving is a natural val-dependent relaxation of being colouration-preserving.

If val is a qualitative valuation associated to $W \subseteq C^\omega$ then a morphism ϕ from G to G' is val-preserving if and only if vertices satisfying W in G are mapped to vertices satisfying W in G'. For simplicity, we say that ϕ is W-preserving in this case.

Universality. Given a class of graphs $\mathcal{C}$ and a graph G, we say that G is *universal for* $\mathcal{C}$ *with respect to* val if every graph of $\mathcal{C}$ has a val-preserving morphism into G. We also say for convenience that G is $(\mathcal{C}, \mathrm{val})$-universal, or simply $\mathcal{C}$-universal when val is clear from context. We say that a graph is *uniformly* val-universal if it is val-universal for the class of all graphs.

2 Well-monotonicity and positionality

2.1 Well-monotonicity and universality

Well-monotonicity. A *well-monotonic* graph $\mathcal{L}$ is a monotonic graph such that $\geqslant$ is a well-order over L. Stated differently $\mathcal{L}$ is monotonic and moreover non-empty sets of vertices have a minimum. A *completely well-monotonic* graph is a well-monotonic graph which is completely monotonic.

Completely well-monotonic graphs can be obtained from well-monotonic graphs simply by adding a $\top$ element.

Lemma 1.2 (Completion of a well-monotonic graph)

Let $\mathcal{L}$ be a well-monotonic graph over L and let $\top \notin L$. Let $\mathcal{L}^\top$ be the graph over $L^\top = L \cup \{\top\}$ where $\top \notin L$ obtained from L by adding all edges from $\top$ to $L^\top$. Then $\mathcal{L}^\top$ is completely well-monotonic and moreover the inclusion $\mathcal{L} \to \mathcal{L}^\top$ is colouration-preserving.

Proof. We extend the well-order from L to $L^\top$ by setting $\top > \ell$ for all $\ell \in L$. It is well known that this produces a complete order over $L^\top$. Given $\ell \in L$ and $c \in C$, c-predecessors in $\mathcal{L}^\top$ are exactly c-predecessor in $\mathcal{L}$ together with $\top$. In particular, every vertex has a c-predecessor in $\mathcal{L}^\top$. There remains to prove monotonicity of $\mathcal{L}^\top$, which follows from monotonicity of the inf-predecessor table.

There is no edge in $\mathcal{L}^\top$ from L to $\top$, hence paths from L are the same in $\mathcal{L}$ and $\mathcal{L}^\top$ and therefore the inclusion is colouration-preserving. □

In a completely well-monotonic graph $\mathcal{L}$ the infimum defining ρ is met therefore we say that ρ is the min-predecessor table of $\mathcal{L}$. It entirely describes the structure of $\mathcal{L}$ since we have

$$\ell \xrightarrow{c} \ell' \text{ in } \mathcal{L} \qquad \Longleftrightarrow \qquad \ell \geqslant \rho(\ell', c).$$

Note that since $\mathcal{L}$ is complete its maximal element $\top$ has c-loops for all c thus any graph has a morphism into $\mathcal{L}$, obtained by mapping all vertices to $\top$. This morphism is of course not val-preserving in general, since the image of any vertex has all colourations.

In general graph-morphisms into a completely well-monotonic graph can be rephrased as prefixpoints of the backpropagation operator.

Lemma 1.3 (Morphisms in $\mathcal{L}$ are prefixpoints)

Let G be a graph over V, let $\mathcal{L}$ be a completely well-monotonic graph and let $\phi : V \to L$ be a progress measure. Then

$$\phi \text{ defines a graph-morphism} \qquad \Longleftrightarrow \qquad \text{Upd}(\phi) \leqslant \phi.$$

Proof. We have

$$\begin{array}{llll} \phi \text{ defines a graph-morphism} & \Longleftrightarrow & \forall v, c, v', & [v \xrightarrow{c} v' \text{ in } G \implies \phi(v) \xrightarrow{c} \phi(v') \in \mathcal{L}] \\ & \Longleftrightarrow & \forall v, & [v \xrightarrow{c} v' \text{ in } G \implies \phi(v) \geqslant \rho(\phi(v'), c)] \\ & \Longleftrightarrow & \forall v, & \phi(v) \geqslant \sup\limits_{v \xrightarrow{c} v' \text{ in } G} \rho(\phi(v'), c) = \text{Upd}(\phi)(v). \end{array}$$ □

Evaluations. Completely well-monotonic graphs are used to *evaluate* graphs (which often represent Eve-strategies) and more generally arenas. Given an arena G, we use $\psi_G^{\mathcal{L}}$ to denote the least fixpoint of $\text{Upd}_G^{\mathcal{L}}$ which we call the *$\mathcal{L}$-evaluation* of G. We often drop the subscript and/or superscript when G and/or $\mathcal{L}$ are clear from context.

Lemma 1.3 allows to rephrase universality for completely well-monotonic graphs.

Lemma 1.4 (Universality of a completely well-monotonic graph)

Let $\mathcal{L}$ be a completely well-monotonic graph, let $\mathcal{C}$ be a class of graphs and let $\text{val} : C^\omega \to X$ be a valuation. Then

$$\mathcal{L} \text{ is } (\mathcal{C}, \text{val})\text{-universal} \qquad \Longleftrightarrow \qquad \forall G \in \mathcal{C}, \quad \psi_G \text{ preserves val}.$$

Proof. The left-to-right implication is direct. Conversely, a morphism $\phi_G : G \to \mathcal{L}$ is $\geqslant \psi_G$ by Lemma 1.3 and by the Knaster-Tarski theorem, hence if ϕ_G is val-preserving then so is ψ_G. □

2.2 Positionality from universality

Completely well-monotonic graphs provide a robust tool for showing positional determinacy (for Eve) on a given class of graphs $\mathcal{C}$. Below is our main theorem in this chapter. The proof is based on Emerson and Jutla's technique [EJ91], and its presentation by Walukiewicz [Wal96]; a similar proof can also be found in the work of Klarlund [Kla92]. We recall that $\mathcal{C}^{\text{paths}}$ and $\mathcal{C}^{\text{ar}}$ respectively denote the class of subgraphs of path-trees of graphs in $\mathcal{C}$, and the class of arenas whose underlying graph belong to $\mathcal{C}$.

Theorem 1.1 (Positionality from universal graphs)

Let val be a valuation and let $\mathcal{C}$ be a class of graphs. If val has a $(\mathcal{C}^{\text{paths}}, \text{val})$-universal completely well-monotonic graph $\mathcal{L}$, then it is uniformly positionally determined over $\mathcal{C}^{\text{ar}}$.

Two ingredients are needed for the proof. First, we show that any (possibly non-positional) strategy σ from v can be used to define a prefixpoint ϕ satisfying $\text{val}_{\mathcal{L}}(\phi(v)) \leqslant \text{val}(\sigma)$. Second, we show that any prefixpoint ϕ defines a uniform positional strategy σ_ϕ satisfying for all vertices v that $\text{val}(\sigma_{\phi,v}) \leqslant \text{val}_{\mathcal{L}}(\phi(v))$. This implies the theorem: the strategy induced by the $\mathcal{L}$-evaluation ψ of G is positional and optimal since it is the least prefixpoint.

At the level of intuition, the first step uses the well-order to *fold* a non-positional strategy into a prefixpoint, and the second step shows that a prefixpoint defines a positional strategy. We fix a $(\mathcal{C}^{\text{paths}}, \text{val})$-universal completely well-monotonic graph $\mathcal{L}$ and an arena $G \in \mathcal{C}^{\text{ar}}$ over V.

From arbitrary strategies to prefixpoints. Consider a strategy σ for Eve from $v_0 \in V$, and its unfolded graph $G_{\sigma,\text{unfold}} \in \mathcal{C}^{\text{paths}}$. We let $\psi : \Pi_\sigma \to \mathcal{L}$ denote the $\mathcal{L}$-evaluation of $G_{\sigma,\text{unfold}}$. We have

$$\text{val}(\sigma) = \text{val}_{G_{\sigma,\text{unfold}}}(\varepsilon) \geqslant \text{val}_{\mathcal{L}}(\psi(\varepsilon)).$$

where the first equality holds by definition, whereas the second one follows from $\mathcal{C}^{\text{paths}}$-universality of $\mathcal{L}$ and Lemma 1.4 (it is actually an equality, but this is the meaningful inequality).

We let $\phi : V \to L$ be the progress measure defined by

$$\phi(v) = \inf\{\psi(\pi) \mid \pi : v_0 \rightsquigarrow_\sigma v \text{ in } G\}.$$

Note that vertices v which are not reached from v_0 by paths consistent with σ are mapped to $\inf \varnothing = \top$, the maximal element in $\mathcal{L}$. For other vertices however, the infimum defining $\phi(v)$ is a minimum thanks to well-orderedness, which is crucial for the result below.

Lemma 1.5 (First ingredient for Theorem 1.1)

The progress measure ϕ is a prefixpoint of $\text{Upd}_G^{\mathcal{L}}$ satisfying

$$\text{val}_{\mathcal{L}}(\phi(v_0)) \leqslant \text{val}(\sigma).$$

Proof. Since ε defines a path from v_0 to v_0 in G it holds that

$$\phi(v_0) \leqslant \psi(\varepsilon) \text{ in } L,$$

therefore by colouration-monotonicity, $\text{val}_{\mathcal{L}}(\phi(v_0)) \leqslant \text{val}_{\mathcal{L}}(\psi(\varepsilon)) \leqslant \text{val}(\sigma)$.

Let $v \in V$, we aim to prove that $\text{Upd}_G(\phi)(v) \leqslant \phi(v)$. If $\phi(v) = \top$, the maximal element in L, then there is nothing to prove. Otherwise, there exists by well-orderdness of $\mathcal{L}$ a path $\pi : v_0 \rightsquigarrow_\sigma v$ in G satisfying $\phi(v) = \psi(\pi) \in L$ and then there are two similar cases according to the player controlling v in G.

- If $v \in V_{\text{Eve}}$, then π has a unique successor in $G_{\sigma,\text{unfold}}$, namely $\pi' = \pi\sigma(\pi)$, and we let $(v \xrightarrow{c_0} v_0') = \sigma(\pi)$ which is an edge in G. Then we have

  $$\begin{aligned}\mathrm{Upd}_G(\phi)(v) = \min_{v \xrightarrow{c} v' \text{ in } G} \rho(\phi(v'), c) \leqslant \rho(\phi(v_0'), c_0) &\overset{(*)}{\leqslant} \rho(\psi(\pi'), c_0) \\ &= \mathrm{Upd}_{G_{\sigma,\text{unfold}}}(\psi)(\pi) = \psi(\pi) = \phi(v),\end{aligned}$$

  where the marked inequality follows from the fact that $\pi' : v_0 \rightsquigarrow v_0'$ in G and therefore $\phi(v_0') \leqslant \psi(\pi')$ by definition of ϕ, together with monotonicity of ρ.

- If $v \in V_{\text{Adam}}$, then $\pi \xrightarrow{c} \pi'$ in $G_{\sigma,\text{unfold}}$ if and only if $\pi' = \pi(v \xrightarrow{c} v')$ with $v \xrightarrow{c} v'$ in G. Thus we now obtain

  $$\mathrm{Upd}_G(\phi)(v) = \sup_{v \xrightarrow{c} v' \text{ in } G} \rho(\phi(v'), c) \leqslant \sup_{v \xrightarrow{c} v' \text{ in } G} \rho(\psi(\pi(v, c, v'), c) = \mathrm{Upd}_{G_{\sigma,\text{unfold}}}(\psi)(\pi) = \phi(v),$$

  concluding the proof. □

As a rephrasal of Lemma 1.5 via Lemma 1.3, we also have the following useful result, which is not formally required for the proof of Theorem 1.1.

Corollary 1.1 (From $\mathcal{C}^{\text{paths}}$ to $\mathcal{C}$)

Let $\mathcal{C}$ be a class of graphs, let $\mathcal{L}$ be a completely well-monotonic graph and let val be a valuation. If $\mathcal{L}$ is $(\mathcal{C}^{\text{paths}}, \text{val})$-universal then it is $(\mathcal{C}, \text{val})$-universal.

From prefixpoint to positional strategy. We now consider a prefixpoint ϕ of Upd_G. For all $v \in V_{\text{Eve}}$ we have

$$\mathrm{Upd}_G(\phi)(v) = \min_{v \xrightarrow{c} v'} \rho(\phi(v'), c) \leqslant \phi(v).$$

We say that a uniform positional strategy $\sigma : V_{\text{Eve}} \to E$ *respects* ϕ if for each $v \in V_{\text{Eve}}$ the edge $\sigma(v) = v \xrightarrow{c} v'$ meets the above minimum.

Lemma 1.6 (Second ingredient for Theorem 1.1)

Assume that σ respects ϕ and let $\pi : v \overset{w}{\rightsquigarrow}_\sigma v'$ be a finite path consistent with σ in G. Then $\phi(v) \overset{w}{\rightsquigarrow} \phi(v')$ in $\mathcal{L}$ and therefore for all $v \in V$ we have

$$\mathrm{val}(\sigma_v) \leqslant \mathrm{val}_{\mathcal{L}}(\phi(v)).$$

Note that the proof below does not require well-foundedness.

Proof. We prove the first statement by induction on the length of π. For paths of length 0 there is nothing to prove. Now let π_G be a path consistent with σ in G of length $\geqslant 1$ and assume the result known for shorter paths.

We write $\pi_G : v \overset{w'}{\rightsquigarrow}_\sigma v' \xrightarrow{c} v''$ and by induction we have $\pi'_{\mathcal{L}} : \phi(v) \overset{w'}{\rightsquigarrow} \phi(v')$ in $\mathcal{L}$. We show that $\phi(v') \xrightarrow{c} \phi(v'')$ in $\mathcal{L}$.

- If $v' \in V_{\text{Eve}}$ then $(v', c, v'') = \sigma(v')$ thus $\rho(\phi(v''), c) \leqslant \phi(v')$ and therefore $\phi(v') \xrightarrow{c} \phi(v'')$.

- If $v' \in V_{\text{Adam}}$ then we have

$$\rho(\phi(v''), c) \leqslant \sup_{v' \xrightarrow{c} u''} \rho(\phi(u''), c) = \text{Upd}_G(\phi)(v') \leqslant \phi(v'),$$

and again the result follows, concluding the induction.

By a transfinite step this proves that colourations of infinite paths from v consistent with σ in G are colourations in $\mathcal{L}$ from $\phi(v)$, thus

$$\text{val}(\sigma_v) = \sup_{v \overset{w}{\rightsquigarrow} \text{ in } G} \text{val}(w) \leqslant \sup_{\phi(v) \overset{w}{\rightsquigarrow} \text{ in } \mathcal{L}} \text{val}(w) = \text{val}_{\mathcal{L}}(\phi(v)).$$ □

This concludes the proof of Theorem 1.1: combining the two lemmas, any strategy can be folded into a positional one whose value is not greater.

3 Prefix-invariance properties and universality

We discuss some generalities about monotonic graphs and universality in the case of objectives with prefix-invariance properties.

3.1 Prefix-increasing objectives

The following result is sometimes useful. It can be seen as being analogous to Lemma 4 (see preliminaries), which states that in the prefix-increasing case positional strategies can always be chosen to be uniform.

Lemma 1.7 (Graph valuations in prefix-increasing case)

Assume that val is prefix-increasing and consider a graph G over V. If two vertices v and v' satisfy $\text{val}_G(v) < \text{val}_G(v')$ then there is no edge in G from v to v'.

Proof. By contradiction, let $e = v \xrightarrow{c} v'$ be an edge in G and pick a path π' from v' with $\text{val}_G(\pi') > \text{val}_G(v)$. Then $e\pi'$ is a path from v and we have

$$\text{val}(e\pi') \geqslant \text{val}(\pi') > \text{val}(v),$$

which is a contradiction since $e\pi'$ is a path from v in G. □

We now consider the case of a prefix-increasing qualitative valuation, given by the objective $W \subseteq C^\omega$, with $W \subseteq cW$ for all colours c. We consider a completely well-monotonic graph which we call $\bar{\mathcal{L}}$, over vertices $\bar{L}$. We let $\mathcal{L}$ be its restriction to the set L of vertices which satisfy W ; by definition, $\mathcal{L}$ satisfies W. The above lemma states in this case that there are no edges in $\bar{\mathcal{L}}$ from L to $\bar{L} \backslash L$.

By colouration monotonicity in $\bar{\mathcal{L}}$, L is a downward-closed subset of $\bar{\mathcal{L}}$, thus $\leqslant$ is a well-ordering over L. Therefore $\mathcal{L}$ is a well-monotonic graph, which is not complete in general. Recall from Lemma 1.2 that $\mathcal{L}^\top$ defines a completely well-monotonic graph obtained from $\mathcal{L}$ by adding a $\top$-element to $\mathcal{L}$ with all outgoing edges. The three monotonic graphs are depicted on Figure 1.3.

Lemma 1.8 (Universality for prefix-increasing W)

Let $\mathcal{C}$ be a class of graphs, and let $\mathcal{C}_W$ be the class of all graphs in $\mathcal{C}$ which satisfy W. The following are equivalent.

(i) $\bar{\mathcal{L}}$ is $(\mathcal{C}, W)$-universal,

(ii) $\mathcal{L}^\top$ is $(\mathcal{C}, W)$-universal,

(iii) $\mathcal{L}$ embeds all graphs from $\mathcal{C}_W$.

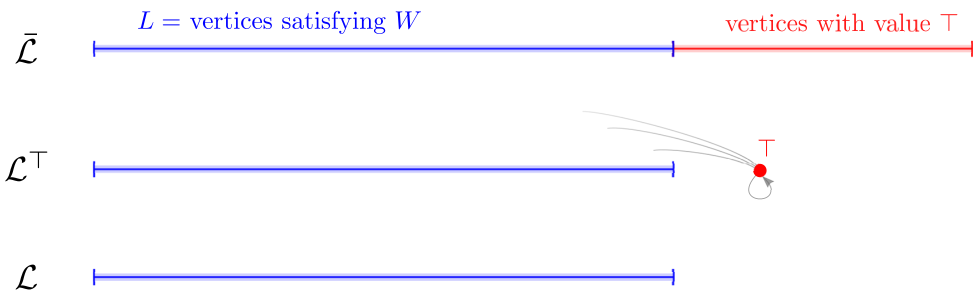


Figure 1.3: The three monotonic graphs in Lemma 1.8. Since there is no edge going from L to its complement in $\bar{L}$, one can safely shrink together vertices with value $\top$ (and add more outgoing edges if needed), leading to the completion $\mathcal{L}^\top$ of $\mathcal{L}$, which in turn carries no more information than $\mathcal{L}$.

Proof. We show that (iii) $\implies$ (ii) $\implies$ (i) $\implies$ (iii) in this order.

Given a graph $G \in \mathcal{C}$ over V, we let V_W denote the set of vertices which satisfy W. By Lemma 1.7, there is no edge in G from V_W to $G \backslash V_W$, hence the restriction G_W of G to W is a graph, and by definition it belongs to $\mathcal{C}_W$. Since there are all edges from $\top$ to L in $\mathcal{L}^\top$, a morphism $\phi : V_W \to L$ extends to a morphism $\phi^\top : V \to L^\top$ by setting $\phi^\top(v) = \top$ for $v \notin V_W$. It is W-preserving by definition: if v satisfies W then $v \in V_W$ thus $\phi^\top(v) \in L$ which satisfies W. This gives the first implication.

For $G \in \mathcal{C}$ over V, if $\phi^\top : V \to L^\top$ is a W-preserving morphism, then $\phi^\top$ maps G_W to L and its complement to $\top$. Now the map $L^\top \to \bar{L}$ which coincides with the identity over L and maps $\top$ to the maximal element of $\bar{L}$ is also W-preserving since there are no edges leaving L in $\bar{\mathcal{L}}$, and it is a morphism since $\top$ has all c-loops in $\bar{\mathcal{L}}$. We conclude with the second implication by composition of W-preserving morphism.

For the third implication, it suffices to see that if G satisfies W then a W-preserving morphism in $\bar{\mathcal{L}}$ embeds G in $\mathcal{L}$. □

Therefore, the notion of being universal in the prefix-increasing qualitative case corresponds to the one of Colcombet and Fijalkow [CF18]: we are looking for a well-monotonic $\mathcal{L}$ which needs not be complete, but

(i) satisfies W, and

(ii) embeds all graphs from $\mathcal{C}$ which satisfy W.

By a slight abuse, we will say, in the qualitative prefix-increasing case, that a graph $\mathcal{L}$ is $\mathcal{C}$-universal if the two above conditions are met. This bypasses the need for systematically introducing the completion $\mathcal{L}^\top$. For statements about the cardinality it makes a difference of at most 1 which can generally be ignored.

3.2 Pregraphs and prefix-decreasingness

We now discuss mapping of pregraphs, which we recall are non-necessarily sinkless graphs, and we fix a pregraph G and a valuation $\mathrm{val} : C^\omega \to X$. We use $\bot_X$ to denote the minimal element of X, given by $\bot_X = \inf X = \sup \varnothing$.

Even though it does not technically correspond to an Adam-controlled arena (since it may have sinks), one can define the values over G by the same formula,

$$\mathrm{val}_G(v) = \sup_{v \overset{w}{\rightsquigarrow} \text{ in } G} \mathrm{val}(w),$$

which takes value $\bot_X$ by definition over sinks. Note that finite paths are not taken into account in the supremum; this corresponds to the convention that Adam loses if he ends up in a sink. According to val, this may or may not be satisfactory. For instance if one considers the safety objective over $\{\mathrm{safe}, \mathrm{bad}\}$ one may want that a finite path which contains bad is winning for Adam even if it leads to a sink, but this is not captured by the above definition. However one may very well work with this definition as long as val is prefix-decreasing (which is not the case of the safety objective) which is the object of this discussion.

Given an completely monotonic $\mathcal{L}$, one may likewise define the backpropagation operator, and therefore the evaluation of G as its least fixpoint, by using the same formula

$$\mathrm{Upd}_G^{\mathcal{L}}(\phi)(v) = \sup_{v \overset{c}{\rightsquigarrow} v' \text{ in } G} \rho(\phi(v'), c)$$

which is $\sup \varnothing = \bot_L$ if v is a sink, where $\bot_L$ denotes the minimal element in L.

We now assume that val is prefix-decreasing and that it is non-trivial in the sense that there is $u \in C^\omega$ such that $\mathrm{val}(u) = \bot_X$. If val is qualitative this means that there are paths which are winning for Eve. We also assume that $\mathcal{L}$ has an element such that

$$\mathrm{val}_{\mathcal{L}}(\ell) = \bot_X,$$

which is necessary for instance if there is a graph G with a vertex of value $\bot_X$ with a val-preserving morphism in $\mathcal{L}$. This will always be the case if $\mathcal{L}$ is val-universal for a non-trivial class of graphs. By colouration-monotonicity we therefore have $\mathrm{val}_{\mathcal{L}}(\bot_L) = \bot_X$.

Now since $\mathcal{L}$ is a graph $\bot_L$ has a successor for some colour which we will denote by $c^- \in C$. By right composition we have $\bot_L \xrightarrow{c^-} \bot_L$. Given a pregraph G over V we construct a graph G' over V simply by appending a c^--loop to all sinks. The following relies on prefix-decreasingness of val.

Lemma 1.9

The identity from G to G' is val-preserving.

Proof. Any infinite path in G' which is not in G ultimately cycles in a c^--loop from a sink, and its colouration is therefore of the form $u(c^-)^\omega$. By prefix-decreasingness of val its valuation is thus $\leqslant \mathrm{val}((c^-)^\omega) \leqslant \mathrm{val}_{\mathcal{L}}(\bot_L) = \bot_X$, since there is a path of colouration $(c^-)^\omega$ from $\bot_L$ in $\mathcal{L}$. □

Hence if G' has a val-preserving morphism in $\mathcal{L}$ then so does G (it is actually easy to see that evaluations of G and G' in this case are equal, but we will not use this fact). A consequence which will be useful in Section 4 in the next chapter is that assuming val is prefix-decreasing and $\mathcal{L}$ is val-universal over the class of all graphs of a given cardinality, then it also val-embeds such pregraphs.

Manipulating well-monotonic graphs 2

Our aim in this second chapter is to manipulate well-monotonic graphs, and give some intuition by working on different examples. Here, by default "positional" means "positional over arbitrary arenas".

Section 1 discusses a few basic ω-regular objectives, namely safety game, a variant, reachability games, Büchi and co-Büchi games. For each case, natural constructions are given and discussed. Each time, we also make a digression about existence of uniformly universal graphs (which turns out to be quite rare), and discuss the cardinality growth of the $\mathcal{C}$-universal graphs when $\mathcal{C}$ grows larger (for small infinite classes of graphs). Although this is completely non-essential, we believe that it gives interesting insights about these different objectives (for instance, the growth is more important for Co-Büchi than Büchi objectives).

In Section 2, we focus on payoff valuations. The energy valuation immediately corresponds (almost, by definition) to a well-monotonic graph over ω, and therefore it is positional. The discounted valuation is not positional over arbitrary arenas, but only over those of finite degree. It actually eludes our technique and admits no universal well-monotonic graph, however it does admit a natural universal monotonic graph over $\mathbb{R}$, and the standard argument due to Shapley [Sha53] allows to circumvent the need for well-foundedness. We take this occasion to discuss positionality proofs for mean-payoff games, and provide one for completeness.

In Section 3, we propose two general variants of counter games, establish their positionality and briefly discuss their significance.

Last, in Section 4, we introduce (finite) lexicographical products of arbitrary prefix-independent objectives. The main result is that if the conditions each have universal well-monotonic graphs, then so does their lexicographical product.

1 Basic ω-regular objectives

1.1 Safety games and a variant

Safety games. The safety objective, given over $C = \{\text{safe}, \text{bad}\}$ by

$$\text{Safety} = \{\text{safe}^\omega\},$$

is the simplest in terms of winning strategies: Eve is guaranteed to win as long as she follows a safe-edge which remains in the winning region. Note that it is prefix-increasing, and thus (see Lemma 1.8) we are looking for a well-monotonic graph $\mathcal{L}$ satisfying Safety and which embeds all graphs satisfying Safety.

Now satisfying safety for a graph simply means not having a bad-edge therefore we have the following result.

Lemma 2.1 (Construction for Safety)

The well-monotonic graph comprised of a single vertex with a safe-loop is uniformly Safety-universal.

This proves thanks to Theorem 1.1 that safety games are positionally determined (which of course has much simpler proofs).

A variation. For the sake of studying a simple example with no prefix-invariance property we consider the objective over $C = \{\text{imm}, \text{safe}, \text{bad}\}$ defined by

$$W = \text{imm}\{\text{imm}, \text{safe}\}^{\omega}.$$

In words, Eve should immediately see the colour imm, and then avoid bad forever. Here, $\text{bad} \cdot W \not\subseteq W$ and $W \not\subseteq \text{safe} \cdot W$. Consider the graph $\mathcal{L}$ depicted in Figure 2.1.

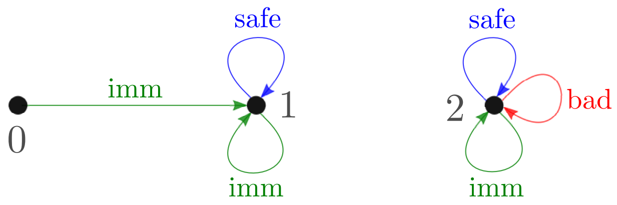


Figure 2.1: A monotonic $\{\text{imm}, \text{safe}, \text{bad}\}$-graph $\mathcal{L}$ over $L = \{0, 1, 2\}$. Edges which follow from composition are not depicted. Note that neither 1 nor 2 satisfy W.

Lemma 2.2

The completely well-monotonic graph $\mathcal{L}$ is uniformly W-universal.

Therefore W is positionally determined over all arenas. Note that in this case several vertices in $\mathcal{L}$ do not satisfy W, and contracting them into one results in losing W-universality since the first would no longer satisfy W. Such a phenomenon is excluded by Lemma 1.8 in the prefix-increasing case.

Proof. Monotonicity of $\mathcal{L}$ is straightforward, order-completeness and well-foundedness are always true for finite sets, and (edge) co-completeness is direct: $\mathcal{L}$ is indeed completely well-monotonic.

Consider any C-graph G over V, and let $V_0, V_1, V_2 \subseteq V$ be the partition of V defined by

- $v \in V_2$ if and only if v has a path which visits a bad-edge, and
- $v \in V_0$ if and only if $v \notin V_2$ and all edges outgoing from v have colour imm.

Note that V_0 is precisely the set of vertices which satisfy W. It is immediate that mapping V_0 to 0, V_1 to 1 and V_2 to 2 defines a W-preserving morphism from G to $\mathcal{L}$. □

1.2 Reachability games

We now consider the reachability objective over $C = \{\text{wait}, \text{good}\}$, given by

$$\text{Reachability} = \{w \in C^\omega \mid |w|_{\text{good}} \geqslant 1\} = C^\omega \backslash \{\text{wait}^\omega\}.$$

A key difficulty. Perhaps surprisingly, constructing universal completely monotonic graphs for the reachability objective turns out to be more involved (and much more interesting) than for the two previous examples: a key difficulty now arises.

Lemma 2.3 (Need for non-uniformity)

There is no graph uniformly Reachability-universal graph.

Proof. Given an ordinal α, we let G_α be the graph over $V_\alpha = \alpha = [0, \alpha)$ given by

$$\lambda \xrightarrow{c} \lambda' \text{ in } G_\alpha \qquad \Longleftrightarrow \qquad c = \text{good or } \lambda > \lambda'.$$

It is illustrated in Figure 2.2. Note that G_α satisfies Reachability by well-foundedness: there is no infinite path of colouration wait^ω.

Assume for contradiction that there exists a Reachability-universal graph G over V for the class of all graphs and let α be an ordinal of greater cardinality $|\alpha| > |V|$. Consider a Reachability-preserving morphism $\phi : G_\alpha \to G$. By our assumption over cardinalities ϕ is not injective and we pick $\lambda > \lambda'$ be such that $\phi(\lambda) = \phi(\lambda')$.

Since $\lambda \xrightarrow{\text{wait}} \lambda'$ in G_λ and ϕ is a morphism, we have $\phi(\lambda) \xrightarrow{\text{wait}} \phi(\lambda') = \phi(\lambda)$ in G. But then $\phi(\lambda) \xrightarrow{\text{wait}} \phi(\lambda) \xrightarrow{\text{wait}} \dots$ defines an infinite path in G which does not satisfy Reachability, contradicting Reachability-preservation of ϕ. □

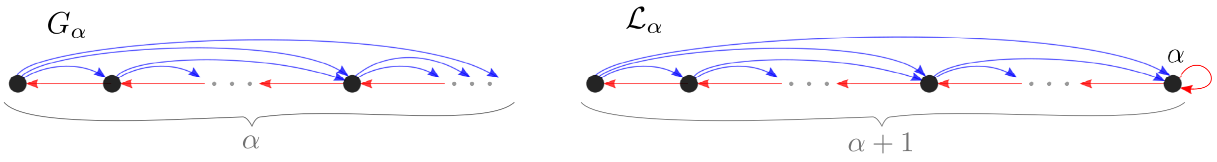


Figure 2.2: The graphs G_α and $\mathcal{L}_\alpha$ (defined below); good-edges are represented in blue and wait edges are red. Some edges which follow by composition are omitted for clarity (for instance, good-edges pointing from right to left), from now on we no longer mention the use of this convention. Note that in $\mathcal{L}_\alpha$, the vertex α does not satisfy Reachability.

Hopefully our notion allows for non-uniform constructions. Note that Reachability is not prefix-increasing therefore elements which do not satisfy the objective in $\mathcal{L}$ may play a non-trivial role.

A non-uniform construction. Given an ordinal α, we let $\mathcal{L}_\alpha$ denote the graph over $L_\alpha = \alpha + 1 = [0, \alpha]$ given by

$$\lambda \xrightarrow{c} \lambda' \text{ in } G_\alpha \qquad \Longleftrightarrow \qquad c = \text{good or } \lambda > \lambda' \text{ or } \lambda = \alpha.$$

Note that $\mathcal{L}_\alpha$ is similar but not identical to the completion $G_\alpha^\top$ of G_α: there are good-edges towards the maximal element.

Lemma 2.4 (Non-uniform construction for Reachability)

For any ordinal α, $\mathcal{L}_\alpha$ is completely well-monotonic and it is Reachability-universal for the class of all graphs of cardinality $< |\alpha|$.

The proof provides a template which will later be adapted to other objectives hence we break it into well-distinguished steps. It explicits the Kleene iteration which defines the evaluation of a graph G in $\mathcal{L}_\alpha$, for a large enough α. This explains the fact that a few steps are generic.

Proof. Monotonicity of $\mathcal{L}_\alpha$ follows from the formulas

$$\rho(\lambda, \text{wait}) = \min(\lambda+1, \alpha) \quad \text{and} \quad \rho(\lambda, \text{good}) = 0.$$

Completeness and well-orderdness are direct, and again by well-foundedness we have

$$\lambda \text{ satisfies Reachability in } \mathcal{L}_\alpha \quad \iff \quad \lambda < \alpha.$$

We now fix an arbitrary graph G over V.

(i) We construct by transfinite recursion an increasing ordinal-indexed sequence of subsets of V by setting for each ordinal λ

$$V_\lambda = \left\{ v \in V \mid v \xrightarrow{c} v' \text{ in } G \implies [c = \text{good or } \exists \beta < \lambda, v' \in V_\beta] \right\}.$$

(ii) We let $U = \bigcup_\lambda V_\lambda$ and prove that if v satisfies Reachability in G then $v \in U$. We proceed by contrapositive and assume that $v_0 \notin U$: for any ordinal λ, $v_0 \notin V_\lambda$. Then v_0 has a wait-edge towards some vertex v_1 such that for all λ, $v_1 \notin V_\lambda$. By a quick induction we build an infinite path $v_0 \xrightarrow{\text{wait}} v_1 \xrightarrow{\text{wait}} \dots$ in G, which guarantees that v_0 does not satisfy Reachability.

(iii) We show that if $V_\lambda = V_{\lambda+1}$ then for all $\lambda' \geqslant \lambda$ we have $V_{\lambda'} = V_\lambda$. This is direct by transfinite induction: assume the result known for all β such that $\lambda \leqslant \beta < \lambda'$ and let $v \in V_\lambda$.

Then any edge from v is either a good-edge or points towards $v' \in L_\beta$ for some $\beta < \lambda'$, and the result follows since $V_\beta \subseteq V_\lambda$.

(iv) We now let α be such that $|\alpha| > |V|$ and prove that $V_\lambda = V_{\lambda+1}$ for some $\lambda < \alpha$. Indeed, if this were not the case, then any map (obtained using the axiom of choice)

$$\begin{array}{rcl} \alpha & \to & V \\ \lambda & \mapsto & v \in V_{\lambda+1} \setminus V_\lambda \end{array}$$

would be injective, a contradiction.

(v) Therefore $U = \bigcup_{\lambda<\alpha} V_\alpha$ and we let $\phi : V \to L_\alpha = [0, \alpha]$ be given by

$$\phi(v) = \begin{cases} \min\{\lambda \mid v \in V_\lambda\} & \text{if } v \in U \\ \alpha & \text{if } v \notin U. \end{cases}$$

By the second item and since λ satisfies Reachability provided it is $< \alpha$, it holds that ϕ preserves Reachability.

(vi) We verify that ϕ defines a graph-morphism, which follows from the definitions of V_λ and of $\mathcal{L}_\alpha$. First, good-edges are preserved independentely of ϕ since they all belong to $\mathcal{L}_\alpha$. Second, wait-edges from ${}^cU$ are preserved since α has all outgoing wait-edges in $\mathcal{L}_\alpha$. Third if $v \xrightarrow{\text{wait}} v'$ is such that $v \in U$ then $\phi(v') < \phi(v)$ by definition of ϕ thus $\phi(v) \xrightarrow{\text{wait}} \phi(v')$. □

Recovering uniformity over finite degree graphs. We finish our study of the reachability condition with a quick side-result of independent interest.

Lemma 2.5 (Uniform Reachability-universality for finite degree graphs)

The completely well-monotonic graph $\mathcal{L}_\omega$ is Reachability-universal for all graphs of finite degree.

Proof. Let $\mathcal{C}$ be the class of graphs of finite degree and $\mathcal{C}^{\text{acyclic}}$ be its restriction to acyclic graphs. We have

$$\mathcal{C}^{\text{paths}} \subseteq \mathcal{C}^{\text{acyclic}} \subseteq \mathcal{C},$$

thus $\mathcal{C}^{\text{acyclic}}$-universality implies $\mathcal{C}^{\text{paths}}$-universality, which implies $\mathcal{C}$-universality by Corollary 1.1.

Let G be an acyclic graph of finite degree over V and let $v \in V$ be a vertex satisfying Reachability. Consider the tree T rooted at v obtained by restricting G to vertices v' such that v has a wait-path to v'. Then T has finite degree and no infinite paths, it is therefore finite by König's lemma.

Consider the ordinal sequence $V_0 \subseteq \cdots \subseteq V_\alpha \subseteq V_{\alpha+1} \subseteq \dots$ from the proof of Lemma 2.4. By a direct induction, a vertex of height h in T belongs to V_h, which concludes. □

1.3 Büchi games

The Büchi condition is defined over the same set of colours $C = \{\text{wait}, \text{good}\}$ by

$$\text{Büchi} = \{w \in C^\omega \mid |w|_{\text{good}} = \infty\}.$$

It is prefix-independent so we aim to construct (non-necessarily completely) well-monotonic graphs which satisfy Büchi and embed graphs satisfying Büchi.

Non-uniform construction. Given an ordinal α, we consider the graph $\mathcal{L}_\alpha$ over $L_\alpha = \alpha = [0, \alpha)$ given by

$$\lambda \xrightarrow{c} \lambda' \text{ in } \mathcal{L}_\alpha \qquad \iff \qquad c = \text{good or } \lambda > \lambda'.$$

Note that this graph is identical to the graph G_α defined in the context of reachability games, and thus we refer to Figure 2.2.

The difference between the completion $(\mathcal{L}_\alpha)^\top$ of the graph defined just above for Büchi and the graph $\mathcal{L}^{\text{R}}$ we used for Reachability is that in the latter there are good-edges towards the maximal element. This reflects the fact that in a reachability game there may be good-edges from the winning region to its complement, which is of course false in a Büchi-game (precisely because they are prefix-independent).

It is a direct check that $\mathcal{L}_\alpha$ is a well-monotonic and that it satisfies Büchi.

Lemma 2.6 (Non-uniform construction for Büchi)

For any ordinal α, $\mathcal{L}_\alpha$ is Büchi-universal for the class of all graphs of cardinality $< |\alpha|$.

We follow the same steps as those of the proof of Lemma 2.4.

Proof. Fix a graph G over V which satisfies Büchi.

(i) We construct by transfinite recursion an ordinal-indexed increasing sequence of subsets of V by the formula

$$V_\lambda = \left\{v \in V \mid v \xrightarrow{c} v' \implies [c = \text{good or } \exists \beta < \lambda, v' \in V_\beta]\right\}.$$

Note that the definition is identical to that of the proof of Lemma 2.4, thus we may skip a few steps below which were already proved.

(ii) We let $U = \bigcup_\lambda V_\lambda$ and prove that $U = V$: from $v_0 \notin U$, we may construct a path $v_0 \xrightarrow{\text{wait}} v_1 \xrightarrow{\text{wait}} \dots$ in G, which contradicts the fact that G satisfies Büchi.

(iii) It again holds that $V_\lambda = V_{\lambda+1}$ implies $V_{\lambda'} = V_\lambda$ for $\lambda' > \lambda$.

(iv) We let α such that $|\alpha| > |V|$ and we have $V_\lambda = V_{\lambda+1}$ for some $\lambda < \alpha$.

(v) Therefore $U = \bigcup_{\lambda<\alpha} V_\alpha = V$ and we let $\phi : V \to L_\alpha = [0, \alpha)$ be given by $\phi(v) = \min\{\lambda \mid v \in V_\lambda\}$.

(vi) We verify that ϕ defines a graph morphism, which follows directly from the definitions. □

Uniformity for Büchi games. Regarding uniform constructions, the proofs of Lemmas 2.3 and 2.5 are very easily adapted to the Büchi objective.

Lemma 2.7 (Uniformity for Büchi)

There is no graph which is Büchi-universal for the class of all graphs. However, $\mathcal{L}_\omega$ is Büchi-universal for the class of all graphs of finite degree.

Proof. For the first statement, we have seen that if $\mathcal{L}_\alpha^{\mathrm{R}} = G_\alpha$ embeds into some graph G of cardinality $< |\alpha|$ then G has a wait$^\omega$-path.

For the second statement, we may again reduce to universality for acyclic graphs of finite degrees. In such a graph G satisfying Büchi and given a vertex v, we again consider the tree comprised of vertices v' with a wait-path from v. It is finite thanks to König's lemma which concludes. □

1.4 Co-Büchi games

Recall the co-Büchi condition over $C = \{\text{safe}, \text{bad}\}$ given by

$$\text{Co-Büchi} = \{w \in C^\omega \mid |w|_{\text{bad}} < \infty\}.$$

Non-uniform construction. It is prefix-independent, thus we aim to construct well-monotonic graphs which satisfy Co-Büchi and embed graphs satisfying Co-Büchi. Given an ordinal α consider the graph $\mathcal{L}_\alpha$ given over $L_\alpha = \alpha = [0, \alpha)$ by

$$\lambda \xrightarrow{c} \lambda' \text{ in } \mathcal{L}_\alpha \quad \iff \quad \begin{array}{llll} c = \text{bad} & \text{and} & \lambda > \lambda' & \text{or} \\ c = \text{safe} & \text{and} & \lambda \geqslant \lambda'. & \end{array}$$

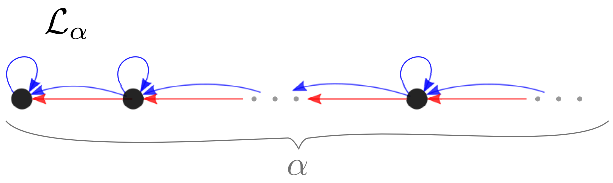


Figure 2.3: The {safe, bad}-graph $\mathcal{L}_\alpha$ defined with respect to the co-Büchi condition; safe-edges are represented in blue and bad edges are red.

It is straightforward to verify that $\mathcal{L}_\alpha$ is well-monotonic and satisfies Co-Büchi.

Lemma 2.8 (Non-uniform construction for co-Büchi)

For any ordinal α, $\mathcal{L}_\alpha$ is Co-Büchi-universal for the class of all graphs of cardinality $< |\alpha|$.

We follow the now familiar template introduced for reachability games.

Proof. Fix a graph G over V which is assumed to satisfy Co-Büchi.

(i) We construct by transfinite induction an ordinal-indexed increasing sequence of subsets of V by the formula

$$V_\lambda = \{v \in V \mid v \overset{\text{safe}^*\text{bad}}{\rightsquigarrow} v' \text{ in } G \implies \exists \beta < \lambda, v' \in V_\beta\}.$$

(ii) We let $U = \bigcup_\lambda V_\lambda$ and prove that $U = V$. Assume that $v_0 \notin U$: for any ordinal λ, $v_0 \notin V_\lambda$. Then v_0 has a safe*bad-path towards some vertex v_1 such that for all λ, $v_1 \notin V_\lambda$. By a quick induction we build an infinite path $v_0 \overset{\text{safe}^*\text{bad}}{\rightsquigarrow} v_1 \overset{\text{safe}^*\text{bad}}{\rightsquigarrow} \dots$ in G, which guarantees that v_0 does not satisfy Co-Büchi, a contradiction.

(iii) We show that if $V_\lambda = V_{\lambda+1}$ then for all $\lambda' \geqslant \lambda$ we have $V_{\lambda'} = V_\lambda$. Again this is direct by transfinite inducion.

(iv) We let α be such that $|\alpha| > |V|$, and again we have $V_\lambda = V_{\lambda+1}$ for some $\lambda < \alpha$.

(v) Therefore $U = \bigcup_{\lambda<\alpha} V_\alpha = V$ and we let $\phi : V \to L_\alpha = [0, \alpha)$ be given by $\phi(v) = \min\{\lambda \mid v \in V_\lambda\}$.

(vi) We verify that ϕ defines a graph-morphism which is direct from the definitions of V_λ and $\mathcal{L}_\alpha$. □

Uniformity for Co-Büchi. Regarding uniform universal graphs, the situation is still the same when considering the class of all graphs but not in case of smaller degrees.

Lemma 2.9 (Uniformity for Co-Büchi)

There is no graph which is uniformly Co-Büchi-universal and $\mathcal{L}_\omega$ is not Co-Büchi-universal for the class of graphs of bounded degree. However $\mathcal{L}_{\omega_1}$ is Co-Büchi-universal for the class of all graphs of countable degree, were ω_1 denotes the first uncountable ordinal.

The fact that there exists a well-monotonic graph of cardinality $\aleph_1$ which is universal for graphs of countable degree is actually general and follows from the forthcoming structuration results of Chapter 3.

Proof. The first statement is easily adapted from the proof of Lemma 2.3, we give the details for completeness. Let G be a graph over V which we assume to be Co-Büchi-universal for the class of all graphs. Let α be an ordinal with cardinality $> |V|$, and consider an embedding ϕ of $\mathcal{L}_\alpha$ into G. It cannot be injective, and we let $\lambda > \lambda'$ in L_α be such that $\phi(\lambda) = \phi(\lambda')$. We have $\lambda \xrightarrow{\text{bad}} \lambda'$ in $\mathcal{L}_\alpha$ thus $\phi(\lambda) \xrightarrow{\text{bad}} \phi(\lambda') = \phi(\lambda)$ in G, which contradicts the fact that G satisfies Co-Büchi. Note that the family of graphs used for the lower bound (namely, $\mathcal{L}_\alpha$) has unbounded (and even infinite) degree.

To show that $\mathcal{L}_{\omega_1}$ is Co-Büchi-universal for all graphs of countable degree it suffices by Corollary 1.1 it suffices to prove the result for subgraphs of paths-graphs of countable degree. Now such a graph is countable, thus if it satisfies Co-Büchi then it embeds in $\mathcal{L}_{\omega_1}$ since $|\omega_1|$ is uncountable.

There remains to see that $\mathcal{L}_\omega$ is not Co-Büchi-universal for the class of all graphs of bounded degree. Consider the graph G over elements of the form $n\uparrow$ and $n\downarrow$ where $n \in \omega$ and given by exactly the edges

$$n\uparrow \xrightarrow{\text{safe}} (n+1)\uparrow, \qquad n\uparrow \xrightarrow{\text{safe}} n\downarrow, \qquad (n+1)\downarrow \xrightarrow{\text{bad}} n\downarrow \qquad \text{and} \qquad 0\downarrow \xrightarrow{\text{safe}} 0\downarrow\ .$$

See Figure 2.4 for an illustration.

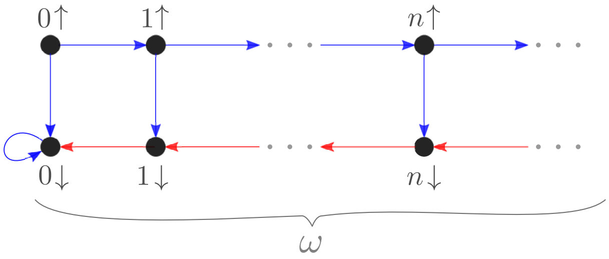


Figure 2.4: The {safe, bad}-graph G for the second lower bound in Lemma 2.9.

It is immediate that G is countable, has degree 2 and satisfies Co-Büchi. Let $\phi : V \to L_{\omega_1}$ be the morphism constructed in the proof of Lemma 2.6. By a direct induction, we have $\phi(n\downarrow) = n$. Thus we obtain $\phi(n\uparrow) = \omega$ for all n. Stated differently there is no embedding of G in $\mathcal{L}_\omega$. □

With some more effort we may generalize the above lower bound to any countable graph.

Lemma 2.10 (No countable Co-Büchi-universal graph for bounded degree)

There is no countable graph which is Co-Büchi-universal for the class of all graphs of bounded degree.

Therefore $\mathcal{L}_{\omega_1}$ is the first graph in the family to be Co-Büchi-universal for the class of all graphs of bounded degree (and it is even universal for the much larger class of graphs of countable degree).

Proof. Let G be a countable graph over V which satisfies Co-Büchi and embeds all bounded degree graphs satisfying Co-Büchi. Consider the ordinal-indexed sequence of subsets V_λ of V constructed in the proof of Lemma 2.8. By item (iv) in the proof, and since $|\omega_1| > |V|$, there is some $\alpha < \omega_1$ such that $V_\alpha = V_\alpha + 1$, or stated differently G embeds in $\mathcal{L}_{\alpha+1}$. Note that $\alpha + 1 < \omega_1$ as it is countable.

Fix a bijection $e : \omega \to \alpha+2 = [0, \alpha+1]$, and consider the graph H over vertices in $(\alpha+2) \times \omega$ given by

$$\forall \lambda, n, \qquad (\lambda, n) \xrightarrow{\text{safe}} (\lambda, n+1) \quad \text{and} \quad e(n) < \lambda \implies (\lambda, n) \xrightarrow{\text{bad}} (e(n), 0).$$

Note that H has degree 2.

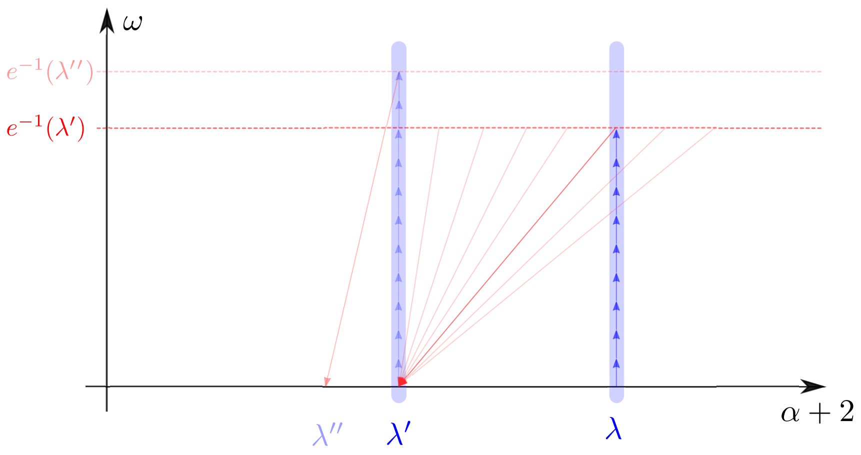


Figure 2.5: The graph H in the proof of Lemma 2.10.

It satisfies Co-Büchi since whenever a bad-edge is seen, the first coordinate decreases, which can happen only finitely many times since it never increases. Now for each $\lambda, \lambda' \in \alpha + 2$, with $\lambda > \lambda'$,

$$(\lambda, 0) \xrightarrow{\text{safe}} (\lambda, 1) \xrightarrow{\text{safe}} \dots \xrightarrow{\text{safe}} (\lambda, e^{-1}(\lambda')) \xrightarrow{\text{bad}} (\lambda', 0)$$

defines a path with colouration in safe*bad from $(\lambda, 0)$ to $(\lambda', 0)$ (it is represented in Figure 2.5). It follows that

$$\lambda > \lambda' \implies (\lambda, 0) \notin V_{\lambda'}$$

and thus H does not embed in $\mathcal{L}_{\alpha+1}$, contradicting the fact that H embeds in G. □

Natural objectives to study next would be parity objectives. However we prefer to present the constructions relative to parity games as obtained by generic lexicographic combinations of Büchi or co-Büchi conditions, which is why we now move to quantitative valuations.

2 Payoff valuations

We now discuss constructions for energy, discounted, and mean-payoff games.

2.1 Energy games

Recall the energy valuation over $C = \mathbb{Z}$ defined by

$$\text{Energy}^+(t_0 t_1 \dots) = \sup_k \sum_{i=0}^{k-1} t_i \in [0, \infty],$$

Consider the graph $\mathcal{L}$ over $L = \omega$ given by

$$\ell \xrightarrow{t} \ell' \text{ in } \mathcal{L} \qquad \iff \qquad t \leqslant \ell - \ell' \in \mathbb{Z}.$$

Note that only non-positive weights are outgoing from 0 in $\mathcal{L}$. See Figure 2.6.

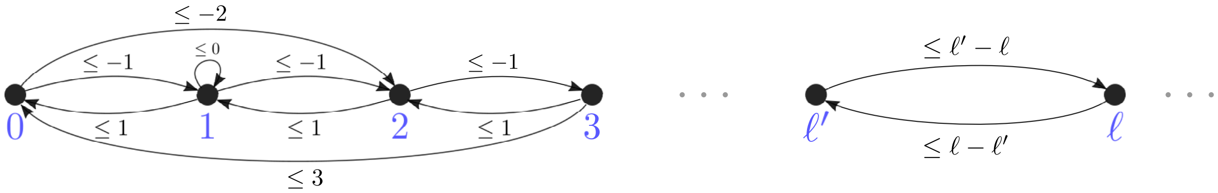


Figure 2.6: The monotonic $\mathbb{Z}$-graph $\mathcal{L}$ corresponding to the Energy$^+$ valuation. The names of the vertices are displayed in blue to improve readability. Not all edges are depicted, we simply write $\xrightarrow{\leqslant t}$ for the conjunction of $\xrightarrow{t'}$ for all $t' \leqslant t$.

The usual order defines a well-order over L and we have

$$\ell \xrightarrow{t} \ell' \text{ in } \mathcal{L} \iff \ell \geqslant \max(0, \ell' + t)$$

thus the min-predecessor table is defined and given by

$$\rho(\ell', t) = \max(0, \ell' + t)$$

therefore $\mathcal{L}$ is well-monotonic. For each $\ell \in \omega$ the path $\ell \xrightarrow{\ell} 0 \xrightarrow{0} 0 \xrightarrow{0} \dots$ has value ℓ, therefore $\text{Energy}^+_{\mathcal{L}}(\ell) \geqslant \ell$.

Conversely consider an infinite path from $\ell_0 \in \omega$. It is of the form $\ell_0 \xrightarrow{t_0} \ell_1 \xrightarrow{t_1} \ell_2 \xrightarrow{t_2} \dots$ with for all i, $t_i \leqslant \ell_i - \ell_{i+1}$. Hence its profiles define a telescoping sum and we have for all n,

$$\sum_{i=0}^{n-1} t_i \leqslant \ell_0 - \ell_n \leqslant \ell_0.$$

Therefore it holds that for all $\ell \in \omega$ we have

$$\text{Energy}^+_{\mathcal{L}}(\ell) = \ell.$$

Energy games are similar to safety games in the sense that they have a uniformly universal well-monotonic graph.

Lemma 2.11 (Uniform construction for energy games)

The completely well-monotonic graph $\mathcal{L}^\top$ is Energy^+-universal for the class of all graphs.

Proof. Consider a graph G over V. We see the values in G as defining a map from V into $L^\top$, formally

$$\begin{array}{rccl} \text{Energy}^+_G : & V & \to & L^\top \\ & v & \mapsto & \text{Energy}^+_G(v), \end{array}$$

where we identify $\top$ to ∞.

The fact that it is Energy^+-preserving follows from the fact that $\text{Energy}^+_{\mathcal{L}}(\ell) = \ell$, proven above. We prove that it is a morphism: consider an edge $e = v \xrightarrow{t} v'$ in G. If $\text{Energy}^+_G(v') = \top$ then $\text{Energy}^+_G(v) \xrightarrow{t} \text{Energy}^+_G(v')$ in $\mathcal{L}^\top$ since $\top$ has all predecessors.

We assume otherwise and let π' a path from v' in G with maximal value $\text{Energy}^+(\pi') = \text{Energy}^+_G(v')$ which we denote by $x' \in \omega$ for simplicity. Then $e\pi'$ defines a path from v in G, therefore

$$\text{Energy}^+_G(v) \geqslant \text{Energy}^+(e\pi') = \max(0, t + x') \geqslant t + x',$$

which rewrites as

$$t \leqslant \text{Energy}^+_G(v) - \text{Energy}^+_G(v'),$$

the wanted result. □

This implies thanks to Theorem 1.1 that energy games over arbitrary arenas are positionally determined. Somewhat ironically, it appears that this result had not been formally established before, Lemma 10 in [BFL+08] is stated over finite arenas[1], whereas Corollary 8 in [CFH14] applies only to arenas of finite degree.

Recall however that the opponent in an energy game can require arbitrary memory even over countable arenas of degree 2 and with bounded weights (see Figure 14). It is well known that no memory is required for the opponent over finite arenas (energy games are bi-positional), this can easily be proved using the one-to-two player lift of Gimbert and Zielonka [GZ05].

2.2 Discounted games

We now consider the discounted valuation over $C = \mathbb{R}$ given by a fixed parameter $\lambda \in (0, 1)$ and defined by

$$\text{Disc}^\lambda(t_0 t_1 \dots) = \sum_{i=0}^{\infty} \lambda^i t_i.$$

We let $\mathcal{L}$ be the graph over $L = \mathbb{R}$ given by

$$\ell \xrightarrow{t} \ell' \text{ in } \mathcal{L} \qquad \iff \qquad t \leqslant \ell - \lambda \ell'$$

Note that $\ell \xrightarrow{\ell} 0 \xrightarrow{0} 0 \xrightarrow{0} 0 \xrightarrow{0} \dots$ defines an infinite path from ℓ in $\mathcal{L}$ with discounted payoff ℓ. Conversely given an infinite path $\ell_0 \xrightarrow{t_0} \ell_1 \xrightarrow{t_1} \dots$ we have

$$\sum_{i=0}^{n-1} \lambda^i t_i \leqslant \sum_{i=0}^{n-1} \lambda^i \ell_i - \lambda^{i+1} \ell_{i+1} = \ell_0 - \lambda^n \ell_{n-1} \xrightarrow[n \to \infty]{} \ell_0,$$

[1] Otherwise, the result would not hold in any case, because it includes the opponent.

and therefore we have for all $\ell \in L$,

$$\mathrm{Disc}^\lambda(\ell) = \ell.$$

Note that $\ell \xrightarrow{t} \ell'$ if and only if $\ell \geqslant t + \lambda\ell'$ therefore $\mathcal{L}$ has minimal predecessors given by

$$\rho(\ell', t) = t + \lambda\ell'.$$

Hence $\mathcal{L}$ is a completely monotonic graph. However it is not well-ordered by $\geqslant$ ($\mathbb{R}$ has infinite decreasing sequences) and therefore Theorem 1.1 cannot be applied; discounted games are of different nature than the ones discussed so far.

However they enjoy a much simpler fixpoint proof of positionality from Shapley [Sha53] which does not require to fold a given non-positional strategy as in Theorem 1.1 (which is where well-foundedness is used). To be more specific $\mathcal{L}$ has a very nice property: operators associated to arenas of finite degree are λ-contracting.

Lemma 2.12 (Operators are contracting)

Let G be an arena of finite degree over V. Then $\mathrm{Upd}_G^{\mathcal{L}}$ is λ-contracting with respect to the infinity norm over $\mathbb{R}^V$, formally

$$||\mathrm{Upd}(\phi_1) - \mathrm{Upd}(\phi_2)|| \leqslant ||\phi_1 - \phi_2||,$$

where $||\phi|| = \max_{v \in V} |\phi(v)|$.

Proof. Let $v \in V_{\mathrm{Eve}}$. We have

$$\begin{aligned} \mathrm{Upd}(\phi_1)(v) - \mathrm{Upd}(\phi_2)(v) &= \min_{v \xrightarrow{t} v'} \left[t + \lambda\phi_1(v')\right] - \min_{v \xrightarrow{t} v'} \left[t + \lambda\phi_2(v')\right] \\ &\leqslant t_2 + \lambda\phi_1(v_2') - t_2 - \lambda\phi_2(v_2') \\ &\leqslant \lambda||\phi_1 - \phi_2||, \end{aligned}$$

where $v \xrightarrow{t_2} v_2'$ minimises $t + \lambda\phi_2(v')$. We obtain $||\mathrm{Upd}(\phi_1)(v) - \mathrm{Upd}(\phi_2)(v)|| \leqslant \lambda||\phi_1 - \phi_2||$ by symmetry and the case of $v \in V_{\mathrm{Adam}}$ is similar. □

Therefore Banach's theorem [Ban22] in the metric space $\mathbb{R}^V$ states that $\mathrm{Upd}_G^{\mathcal{L}}$ has a unique fixpoint, even with arbitrary V. We obtain positionality as a consequence.

Corollary 2.1 (Positionality of finite degree discounted games)

Discounted games of finite degree are positionally determined for both players.

Proof. Consider the unique fixpoint ψ of $\mathrm{Upd}_G^{\mathcal{L}}$, and let σ be a uniform positional strategy which respects ψ. Lemma 1.6 yields for all v

$$\mathrm{Disc}^\lambda(\sigma_v) \leqslant \mathrm{Disc}_{\mathcal{L}}^\lambda(\psi(v)) = \psi(v).$$

Inverting the roles of the two players gives the same operator and therefore the same fixpoint, and thus a uniform positional strategy τ for Adam which respects ψ is such that

$$\mathrm{Disc}^\lambda(\tau_v) \geqslant \psi(v)$$

which implies that $\mathrm{Disc}^\lambda(v) = \psi(v)$ and that it is uniformly reached by both positional strategies σ and τ. □

2.3 Mean-payoff games and equivalence over finite arenas

We have seen that energy and discounted valuations have uniformly universal monotonic graphs (and even well-monotonic in the case of energy games). In this regards mean-payoff games are different, and positionality proofs are less straightforward.

Techniques for proving their bi-positionality over finite arenas are the following (in chronological order).

(i) Ehrenfeucht and Mycielski [EM79] studied interactions with cyclic games (see also [AR17] for extensions), which are over as soon as a cycle is closed.

(ii) Gurvich, Karzanov and Khachiyan [GKK88] observed that positionality follows from existence of *ergodic potentials*, which is a consequence of much more general results of Moulin [Mou76]. They also provided an algorithmic proof, essentially by reduction to energy games (although this terminology was introduced only much later).

(iii) Puri [Pur95] observed that a direct proof can be given using the reduction to discounted games established by Zwick and Paterson [ZP95].

(iv) The one-to-two player lift of Gimbert and Zielonka [GZ05] gives a quick and easy proof.

(v) Mean-payoff games (with $\limsup$ semantic, as ours) are concave therefore Kopczyński's result [Kop06] applies.

For completeness, we will give a proof, and we choose the technique of Puri which is self-contained. This also gives us the occasion to present the reduction to discounted games, and on the way we also formally prove the equivalence with energy games over finite arenas.

The *sum* of a word $w \in \mathbb{Z}^*$ is simply the sum of its letters, and we say that w is negative, non-positive, zero, non-negative or positive if its sum is. As always, this terminology is extended to finite paths by considering their colouration. Here is the key technical result.

Lemma 2.13

Let n and N be natural numbers. There exists $\lambda < 1$ such that for any $w, w' \in [-N, N]_{\mathbb{Z}}^{\leqslant n}$,

$$w' \text{ is positive} \quad \Longrightarrow \quad \mathrm{Disc}^{\lambda}(w(w')^{\omega}) > nN.$$

The choice of nN here is motivated by what follows, but any value can be achieved with λ sufficiently close to 1.

Proof. We denote $w = t_0 \dots t_{r-1}$, $s = t_0 + \dots + t_{r-1}$, $s_\lambda = t_0 + \lambda t_1 + \dots + \lambda^{r-1} t_{r-1}$ and put $w' = t'_0 t'_1 \dots t'_{r'-1}$ and define s' and s'_λ likewise. We assume that $s' > 0$ therefore $s' \geqslant 1$ (and $r' \geqslant 1$). We have

$$|s' - s'_\lambda| = |(1-\lambda)t'_1 + (1-\lambda^2)t'_2 + \dots + (1-\lambda^{r'-1})t'_{r'-1}| \leqslant nN(1-\lambda^{n-1})$$

therefore if λ is close enough to 1 the above is $\leqslant 1/2$ thus $s'_\lambda \geqslant 1/2$ and we have

$$\mathrm{Disc}^{\lambda}(w(w')^{\omega}) = \sum_{i=0}^{r-1} \lambda^i t_i + \lambda^r \sum_{i=0}^{\infty} \lambda^{ir'} s'_\lambda = s_\lambda + \frac{\lambda^r s'_\lambda}{1-\lambda^{r'}} \geqslant -nN + \frac{\lambda^n}{2(1-\lambda^n)}$$

which grows arbitrarily large independently of w or w' when λ tends to 1. □

We now give a very useful result which we will also be refered to in later chapters. Recall that graphs are identified to arenas fully controlled by Adam: their values are suprema over paths.

Lemma 2.14 (Equivalence over finite graphs)

Let G be a finite $[-N,N]_{\mathbb{Z}}$-graph of size n and let v be one of its vertices. The following are equivalent.

(i) All cycles reachable from v are non-positive.

(ii) It holds that $\mathrm{MP}_G(v) \leqslant 0$.

(iii) It holds that $\mathrm{Energy}^+_G(v) < \infty$.

(iv) It holds that $\mathrm{Energy}^+_G(v) \leqslant (n-1)N$.

(v) It holds that $\mathrm{Disc}^\lambda_G(v) \leqslant nN$ where λ is close enough to 1 as given by Lemma 2.13.

Proof. We first show the following chain of implications

$$\neg(i) \xrightarrow{1} \neg(ii) \xrightarrow{2} \neg(iii) \xrightarrow{3} \neg(iv) \xrightarrow{4} \neg(i),$$

and finish with the equivalence $(i) \xrightarrow{5} (v) \xrightarrow{6} (i)$.

1. If $v \overset{w}{\rightsquigarrow} v' \overset{w'}{\rightsquigarrow} v'$ where w' has positive sum then $v \overset{w}{\rightsquigarrow} v' \overset{w'}{\rightsquigarrow} v' \overset{w'}{\rightsquigarrow} \dots$ defines an infinite path from v with mean-payoff $\geqslant 1/|w'| > 0$.
2. Any path of positive mean-payoff has unbounded profile.
3. This is trivial.
4. A path $\pi : v_0 \xrightarrow{t_0} v_1 \xrightarrow{t_1} \dots$ with $\mathrm{Energy}^+(\pi) > (n-1)N$ has by definition $r \in \omega$ such that $\sum_{i=0}^{r-1} t_i > (n-1)N$. Let π_0 be a path with minimal such r. Since all t_i's are $\leqslant N$ it must be that $r \geqslant n$, and therefore there is a repetition in $v_0, v_1, \dots, v_r$. This defines a cycle, which has positive weight by minimality of r since otherwise removing it would produces a shorter path with a greater sum.
5. By positional determinacy for Adam there is a path of the form $v \overset{w}{\rightsquigarrow} v \overset{w'}{\rightsquigarrow} v' \overset{w'}{\rightsquigarrow} \dots$ with $|w| \leqslant n-1$ which has a maximal discounted-payoff. Since w' has non-positive sum we have with the notations of Lemma 2.13 that $s'_\lambda \leqslant 1/2$ therefore $\mathrm{Disc}^\lambda(v) = s_\lambda + \lambda^{|w|} s'_\lambda \leqslant (n-1)N + 1/2$.
6. If $v \overset{w}{\rightsquigarrow} v' \overset{w'}{\rightsquigarrow} v'$ where w' has positive sum then the discounted-payoff of v is larger than that of $w(w')^\omega$ which concludes thanks to Lemma 2.13. □

We are now ready to prove the sought result.

Theorem 2.1 (Positionality of finite threshold mean-payoff games)

The objective $\mathrm{MP}^{\leqslant 0}$ is uniformly positionally determined for both players over finite $[-N,N]_{\mathbb{Z}}$-arenas.

Proof. Let G be such an arena of size n over V and let λ be given by Lemma 2.13. Let σ, τ be optimal uniform positional strategies for each players with respect to Disc^λ in G.

Let $v \in V$ and assume that $\mathrm{Disc}^\lambda_G(v) \leqslant nN$. Applying Lemma 2.14 in $G_{\sigma,v}$ yields $\mathrm{MP}(\sigma_v) \leqslant 0$. Assume conversely that $\mathrm{Disc}^\lambda_G(v) > nN$. Then applying Lemma 2.14 in $G_{\tau,v}$ yields $\mathrm{MP}(\tau_v) > 0$.

Therefore the winning regions of $\mathrm{MP}^{\leqslant 0}$ in G coincide with those of $(\mathrm{Disc}^\lambda)^{\leqslant nN}$, and moreover σ and τ are optimal with respect to $\mathrm{MP}^{\leqslant 0}$. □

This implies positionality of the mean-payoff valuation via a standard reduction.

Corollary 2.2 (Positionality of mean-payoff valuation)

The valuation MP is uniformly positionally determined for both players over finite $\mathbb{Q}$-arenas.

Proof. First observe that positional determinacy of $\mathrm{MP}^{\leqslant 0}$ over finite $\mathbb{Q}$-arenas follows from the theorem simply by multiplying by a common denominator.

Let G be such an arena and let $t \in \mathbb{Q}$. Consider the arena G^t obtained from G by adding t to all weights. A path with mean-payoff x in G corresponds to a path with mean-payoff $x + t$ in G^t.

Now let $v \in V$, let $x \in \mathbb{R}$ denote the mean-payoff value of v in G and let $\varepsilon > 0$ be such that $x + \varepsilon \in \mathbb{Q}$. Then v has mean-payoff value $-\varepsilon \leqslant 0$ in the $\mathbb{Q}$-arena $G_{-x-\varepsilon}$ and therefore there is a positional strategy σ_ϵ from v with value $\leqslant 0$ in $G_{-x-\varepsilon}$ and hence $\leqslant x + \varepsilon$ in G.

Since there are only finitely many positional strategies, one of them achieves value $\leqslant x+\varepsilon$ for ε arbitrarily small therefore it achieves value $\leqslant x$. This gives positional determinacy for Eve, the result for the other player is obtained by symmetry. Uniform positional determinacy in this case follows from prefix-independence as we proved in the preliminaries. □

Besides a positionality proof for finite mean-payoff games via reduction to discounted games, Lemma 2.14 implies the following result which is important for its algorithmic consequences (see Chapter 6). We let $\mathcal{L}$ denote the well-monotonic graph over ω introduced for energy games, and $\mathcal{L}_{[0,(n-1)N]}$ denote its restriction to $[0,(n-1)N]$.

Corollary 2.3 (A small $\mathrm{MP}^{\leqslant 0}$-universal monotonic graph)

The finite well-monotonic graph $\mathcal{L}_{[0,(n-1)N]}$ is $\mathrm{MP}^{\leqslant 0}$-universal for the class of all $[-N,N]$-graphs of cardinality $\leqslant n$.

The proof of the corollary is direct from $(ii) \iff (iv)$ in Lemma 2.14 and the fact that Energy^+_G defines a morphism from G to $\mathcal{L}$ if G satisfies $MP^{\leqslant 0}$, see Lemma 2.11.

3 Counter games

We now discuss two variants of counter games. The set of colours C is the set of all monotonic functions $f : \omega \to \omega$, which are seen as acting on a (non-negative) counter. This makes for quite a general class of games as it includes the possibilities of incrementing, decrementing, setting the counter to any value (including resetting to zero), halving, multiplying, raising n to 2^{2^n} or to the next prime number, or even applying non-computable monotonic functions. We first discuss backward counter games which are less natural but a bit easier as they directly generalise energy games.

3.1 Backward counter games

Consider the *backward counter valuation* given by

$$\text{BackwardSup}(f_0 f_1 \dots) = \sup_n f_0(f_1(\dots(f_n(0))\dots)) \in [0, \infty].$$

It is not hard to see that Energy^+ coincides with the above valuation if each weight t is replaced by the monotonic function $n \mapsto \max(0, n+t)$.

We extend the well-monotonic graph considered for the energy valuation to the current setting by letting $\mathcal{L}$ be the C-graph defined over $L = \omega$ by

$$\ell \xrightarrow{f} \ell' \text{ in } \mathcal{L} \qquad \iff \qquad \ell \geqslant f(\ell').$$

The study of $\mathcal{L}$ is not harder than for the special case of energy games. Monotonicity of $\mathcal{L}$ follows from monotonicity of its min-predecessor tables given by

$$\rho(\ell', f) = f(\ell').$$

Given $n \in \omega$ we let $\bar{n} \in C$ denote the (monotonic) constant n function. Given $\ell \in \omega = L$ the path $\ell \xrightarrow{\bar{\ell}} 0 \xrightarrow{\bar{0}} 0 \xrightarrow{\bar{0}} \dots$ has value $\bar{\ell}(0) = \ell$ therefore the value of ℓ is $\geqslant \ell$.

Conversely given an infinite path $\pi : \ell_0 \xrightarrow{f_0} \ell_1 \xrightarrow{f_1} \dots$, we have by definition for all i that $\ell_i \geqslant f_i(\ell_{i+1})$. Therefore it follows from a direct induction (thanks to monotonicity of the f_i's) that for all n we have

$$\ell_0 \geqslant f_0(f_1(\dots(f_n(\ell_{n+1}))\dots)) \geqslant f_0(f_1(\dots(f_n(0))\dots)),$$

hence by taking a supremum over n we have $\ell_0 \geqslant \text{BackwardSup}(\pi)$ and therefore the value of ℓ_0 is exactly ℓ_0.

Lemma 2.15 (Universal well-monotonic graph for BackwardSup)

The completely well-monotonic graph $\mathcal{L}$ is uniformly BackwardSup-universal.

The proof follows the same lines as that of Lemma 2.11.

Proof. Let G be a C-graph over V and consider the map $\text{BackwardSup}_G : V \to L^\top$ where ∞ is identified with $\top$. It is BackwardSup-preserving as shown above so we are left with proving that it is a morphism.

Let $e = v \xrightarrow{f} v'$ in G. If v' has value ∞ there is nothing to prove since $\top$ has all predecessors in $\mathcal{L}^\top$. Therefore we assume otherwise, let π' define a maximal path from v' in G and let x' denote the value of π' (which is also the value of v'). Then $e\pi'$ defines a path from v in G thus

$$\text{BackwardSup}(v) \geqslant \text{BackwardSup}(e\pi') = f(x'),$$

and hence

$$\text{BackwardSup}(v) \xrightarrow{f} \text{BackwardSup}(v') \in \mathcal{L}^\top. \qquad \square$$

This implies positionality over arbitrary arenas for the backward counter valuation.

A class of valuations. Note that $\mathcal{L}$ has a very strong universality property: for any monotonic C'-graph $\mathcal{L}'$ over $L' = \omega$ with the usual order, there is a renaming $C' \to C$ of the colours such that the identity $L' \to L = \omega$ embeds $\mathcal{L}'$ in $\mathcal{L}$. In other words, any valuation which admits a uniformly universal monotonic graph over (a subset of) ω can be reduced to BackwardSup, which could therefore be called *complete* for this class of valuations. Can this natural class of valuations, which gives a broad generalisation of safety objectives, be better understood, or characterised?

Continuous bi-positional valuations. A surprising parallel can be made with the recent work of Kozachinskiy, who proved (see Theorem 22 in [Koz21a]) that any *continuous* valuation $A^\omega \to \mathbb{R}$ which is bi-positional over *finite* arenas, has a similar form (involving contracting monotonic maps $f : K \to K$ where $K \subseteq \mathbb{R}$ is compact, and replacing sup with lim). In particular, bi-positionality of such valuations can be established in general via existence of a unique fixpoint, as for discounted games (see Section 2.2); it does not hold in general however that these can be reduced to (even multi) discounted games [Koz21a].

3.2 Boundedness games

We now discuss *boundedness games*, which are C-arenas equipped with the objective

$$\mathrm{Bounded}_N = \{w \in C^\omega \mid \forall n, \quad f_n(f_{n-1}(\dots(f_0(0))\dots)) \leqslant N\},$$

where $N \in \omega$ is a fixed bound. In contrast with backward counter games, the maps are now applied in chronological order (first f_0, then f_1 and so on) which corresponds to the natural intuition of updating a counter in place.

Colcombet, Fijalkow and Horn [CFH14] have established that $\mathrm{Bounded}_N$ is positionally determined over arenas of finite degree, we extend this result to arbitrary arenas. Note that $\mathrm{Bounded}_N$ is a prefix-increasing objective by monotonicity of the maps in C, therefore we are looking to construct a well-monotonic graph which satisfies $\mathrm{Bounded}_N$ and embeds such graphs.

We let $\mathcal{L}$ be the graph over $L = [0, N]$ given by

$$\ell \xrightarrow{f} \ell' \text{ in } \mathcal{L} \qquad \iff \qquad f(\ell) \leqslant \ell'.$$

The graph $\mathcal{L}$ is monotonic with respect to the inverse order over $L = [0, N]$, with minimal element N and maximal element 0. It is well-monotonic since all finite orders are well founded. Note that fixing the bound N is required for well foundedness; defining $\mathcal{L}$ over ω as we did before fails when considering the dual ordering.

Theorem 2.2 (Uniform construction for boundedness games)

The graph $\mathcal{L}$ is uniformly $\mathrm{Bounded}_N$-universal.

Note that therefore, boundedness games *with fixed* N belong to the class mentioned above. This however fails if for instance N is quantified existentially.

Proof. We first show that $\mathcal{L}$ satisfies $\mathrm{Bounded}_N$: let $\pi : \ell_0 \xrightarrow{f_0} \ell_1 \xrightarrow{f_1} \dots$ be an infinite path in $\mathcal{L}$. By definition it holds for all i that $f_i(\ell_i) \leqslant \ell_{i+1}$ which implies by monotonicity that for all n,

$$f_n(f_{n-1}(\dots(f_0(0))\dots)) \leqslant \ell_{n+1} \leqslant N,$$

the wanted result.

We define a valuation[2]

$$\begin{array}{rrcl} \mathsf{val}: & C^\omega & \to & [0,N] \cup \{\bot\} \\ & f_0 f_1 \dots & \mapsto & \max\{i \in [0,N] \mid \forall n, \quad f_n(f_{n-1}(\dots(f_0(i))\dots)) \leqslant N\}. \end{array}$$

The (complete) linear order over $[0,N] \cup \{\bot\}$ is again the reverse order, in particular $\bot$ should be thought of as "after zero". For clarity, we still use $\geqslant$, min and max for the usual ordering over integers ; it is understood above that $\max \varnothing = \bot$.

Consider a C-graph G over V which satisfies $\mathsf{Bounded}_N$, we prove that $\mathsf{val}_G : V \to [0,N] = L$ which assigns $\min_{v \overset{w}{\rightsquigarrow}} \mathsf{val}(w)$ to $v \in V$ defines a morphism. Let $e_0 = v_0 \xrightarrow{f_0} v_1$ in G and let $\pi_1 = v_1 \xrightarrow{f_1} v_2 \xrightarrow{f_2} \dots$ be an infinite path from v_1 in G with minimal valuation $i_1 = \mathsf{val}(\pi_1) = \mathsf{val}_G(v_1)$.

Then $\pi_0 = e_0\pi_1$ is a path from v_0 in G thus $\mathsf{val}_G(v) \leqslant \mathsf{val}(\pi_0)$ which we denote by i_0. Note that both i_0 and i_1 are $\geqslant 0$ since G satisfies $\mathsf{Bounded}_N$. We have by definition

$$i_0 = \max\{i \in [0,N] \mid \forall n, f_n(f_{n-1}(\dots(f_0(i))\dots)) \leqslant N\},$$

hence for all n it holds that $f_n(f_{n-1}(\dots(f_0(i_0))\dots)) \leqslant N$. Since

$$i_1 = \max\{i \in [0,N] \mid \forall n, f_n(f_{n-1}(\dots(f_1(i))\dots)) \leqslant N\},$$

we have in particular that $f_0(i_0) \leqslant i_1 = \mathsf{val}_G(v_1)$. By monotonicity of f_0, this implies $f_0(\mathsf{val}_G(v_0)) \leqslant \mathsf{val}_G(v_1)$, thus

$$\mathsf{val}_G(v_0) \xrightarrow{f_0} \mathsf{val}_G(v_1)$$

belongs to $\mathcal{L}$, which concludes the proof. □

4 Lexicographical products

4.1 Definitions

Product of objectives. We assume given two prefix-independent objectives $W_1 \subseteq C_1^\omega$ and $W_2 \subseteq C_2^\omega$, where C_1 and C_2 are disjoint. We let $C = C_1 \sqcup C_2$ and for $w \in C^\omega$ we let $w_1 \in C_1^{\leqslant\omega}$ and $w_2 \in C_2^{\leqslant\omega}$ be the finite or infinite words obtained by restricting w to colours of in C_1 and C_2. Note that if w_2 is finite then w_1 is infinite.

We define the *lexicographical product* of W_1 and W_2 by

$$W = W_1 \otimes W_2 = \left\{ w \in C^\omega \;\middle|\; \begin{array}{l} w_2 \text{ is infinite and } w_2 \in W_2 \text{ or} \\ w_2 \text{ is finite and } w_1 \in W_1 \end{array} \right\}.$$

We stress the fact that this operation is not commutative; intuitively, more importance is given here to W_2. The lexicographical product is however associative, and given three prefix-independent objectives W_1, W_2 and W_3 over disjoint sets of colours we have

$$(W_1 \otimes W_2) \otimes W_3 = W_1 \otimes (W_2 \otimes W_3).$$

[2] It is straightforward to see that $\mathcal{L}^\top$ is in fact universal with respect to this valuation, which is a bit more precise than the statement of the theorem.

More generally, given a *finite* sequence of prefix-independent objectives $W_1, \dots, W_d$ respectively over disjoint $C_1, \dots, C_h$ we let $C = \bigsqcup_{p=1}^{h} C_p$ and define the lexicographical product of $W_1, \dots, W_h$ by

$$\bigotimes_{p=1}^{h} W_p = \{w \in C^\omega \mid w_{p_0} \in W_{p_0} \text{ where } p_0 = \max\{p \mid w_p \text{ is infinite}\}\}.$$

Given $c \in C$ we say that $p \in \{1, \dots, h\}$ is its *priority* if $c \in C_p$, and we let $P = \{1, \dots, h\}$ denote the set of priorities. In a C-graph, we also say for convenience that an edge has priority p if its colour has priority p. Note that given $w \in C^\omega, c \in C$ and $p \in P$ it holds that w_p is infinite if and only if $(cw)_p$ is, and therefore W is prefix-independent.

Since we will later manipulate lexicographical products of more than two objectives we give all definitions and proofs in this context.

Product of monotonic graphs. We now assume given a monotonic graph $\mathcal{L}_p$ over L_p for each $p \in P$ and we let

$$L = \Pi_{p \in P} L_p,$$

be their cartesian product. The order $\geqslant$ over L is defined by lexicographically extending the orders over the L_p's, formally for all $\ell \neq \ell'$,

$$\ell > \ell' \quad \iff \quad \ell_{p_0} > \ell'_{p_0} \text{ where } p_0 = \max\{p \in P \mid \ell_p \neq \ell'_p\}.$$

It is well-known that if the orders over the L_p's are well-orders then so is $\geqslant$.

Given $p_0 \in P$ we also let $L_{\geqslant p_0}$ denote the cartesian product of the L_p's for $p \geqslant p_0$. Just like L, all $L_{\geqslant p_0}$'s ordered lexicographically. Now given $\ell \in L$, we let $\ell_{\geqslant p} \in L_{\geqslant p}$ be obtained via the natural projection. This allows us to define a sequence of preorders $\geqslant_p$ for $p \in P$ over L given by

$$\ell \geqslant_p \ell' \quad \iff \quad \ell_{\geqslant p} \geqslant \ell'_{\geqslant p}.$$

Intuitively, this corresponds to first restricting to the first few most important coordinates, and then comparing lexicographically. These preorders have often been considered in the literature for parity games and were first introduced in this setting by Emerson and Jutla [EJ91] (similar notions were also considered for Rabin games by Klarlund [Kla91; Kla92] and Dexter and Klarlund [KK91]).

Note that $\geqslant_1$ coincides with $\geqslant$ and if $p' \leqslant p$ then $\ell \geqslant_{p'} \ell' \implies \ell \geqslant_p \ell'$: the smaller the index, the finer the preorder. We let $>_p$ and $=_p$ respectively denote the associated strict preorders (which correspond to $\not\leqslant_p$) and equivalence classes; note that $\ell =_p \ell'$ if and only if $\ell_{\geqslant p} = \ell'_{\geqslant p}$.

We now define $\mathcal{L}$ to be the graph over L given by

$$\forall c_p \in C_p, \ell, \ell' \in L, \qquad \ell \xrightarrow{c_p} \ell' \text{ in } \mathcal{L} \quad \iff \quad \ell >_{p+1} \ell' \text{ or } (\ell =_{p+1} \ell' \text{ and } \ell_p \xrightarrow{c_p} \ell'_p \text{ in } \mathcal{L}_p).$$

See Figure 2.7 for an example.

Note that if $\ell \xrightarrow{c_p} \ell'$ then $\ell \geqslant_{p+1} \ell'$ holds in general.

Lemma 2.16

The graph $\mathcal{L}$ is a monotonic.

Proof. Let us verify left composition in $\mathcal{L}$. Let $\ell, \ell', \ell'' \in L$ and let $c_p \in C_p$ be such that

$$\ell \geqslant \ell' \xrightarrow{c_p} \ell'' \text{ in } \mathcal{L}.$$

There are two cases.

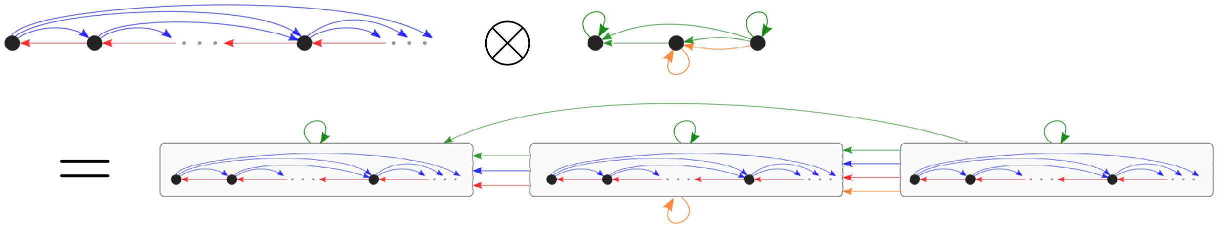


Figure 2.7: A lexicographical product of two monotonic graphs. An edge $B \xrightarrow{c} B'$ between two boxes $B, B' \subseteq V$ depicts the presence of all edges from B to B' (this notational convention is used throughout the thesis).

- If $\ell >_{p+1} \ell''$ then by definition $\ell \xrightarrow{c_p} \ell''$ in $\mathcal{L}$.
- Otherwise we have
$$\ell'' \geqslant_{p+1} \ell \geqslant_{p+1} \ell' \xrightarrow{c_p} \ell'',$$
thus it cannot be that $\ell' >_{p+1} \ell''$, and therefore we have $\ell' =_{p+1} \ell''$, therefore the inequalities above are equivalences, and $\ell'_p \xrightarrow{c_p} \ell''_p \in \mathcal{L}_p$. Since moreover $\ell_p \geqslant \ell'_p$, the wanted result follows by left composition in $\mathcal{L}_p$.

The proof of right composition follows exactly the same lines and we omit it. □

We say that $\mathcal{L}$ is the *lexicographical product of* $\mathcal{L}_1, \dots, \mathcal{L}_h$ and denote it by

$$\mathcal{L} = \bigotimes_{p=1}^{h} \mathcal{L}_p.$$

Again this operation is associative but not commutative. If $\mathcal{L}_1, \dots, \mathcal{L}_p$ are well-monotonic, then so is $\mathcal{L}$.

4.2 Statement of the result and examples

We may now state our main theorem in this section. We recall that prefix-independence of the W_p's is assumed when considering their lexicographical product. We use the notations introduced above.

Theorem 2.3 (Universality of lexicographical product)

Let κ be a cardinal number, and assume that for each $p \in P$, $\mathcal{L}_p$ is well-monotonic and W_p-universal for the class of all C_p-graphs of cardinality $\leqslant \kappa$. Then $\mathcal{L}$ is W-universal for the class of all C-graphs of cardinality $\leqslant \kappa$.

Before going on to the proof, we give a few motivational examples.

Adding a neutral letter. Consider the trivial objective $W_1 = \{0^\omega\}$ over $C = \{0\}$: Eve always wins. It is prefix-independent, and we let W_2 be another prefix-independent objective. Then $W_1 \otimes W_2$ is the objective obtained by adding 0 as a strongly neutral letter (this terminology is introduced in the next chapter).

The graph with a single 0-loop is a well-monotonic graph which is W_1-universal for the class of all graphs and therefore if $\mathcal{L}_2$ is a well-monotonic graph which is W_2-universal for C-graphs of cardinality $\leqslant \kappa$, then so is $\mathcal{L}_1 \otimes \mathcal{L}_2$ for $W_1 \otimes W_2$ and $(C \sqcup \{0\})$-graphs. The lexicographical product $\mathcal{L}_1 \otimes \mathcal{L}_2$ is simply obtained by appending 0-edges to $\mathcal{L}_2$ such that $\xrightarrow{0}$ coincides with $\geqslant$.

Parity games. Let $P = \{1, \dots, h\}$ and for each $p \in P$ we let W_p be the (prefix-independent) co-Büchi objective over $C_p = \{2p, 2p+1\}$ given by

$$W_p = \{w \in C_p^\omega \mid |w|_{2p+1} < \infty\}.$$

Their lexicographical product $W = \bigotimes_{p=1}^h W_p$ is given by

$$W = \big\{w \in [1, 2h]^\omega \mid |w|_{2p_0+1} < \infty \text{ where } p_0 = \max\{p \in P \mid w_p \text{ is infinite}\}\big\},$$

which coincides with the parity objective over $C = [1, 2h]$. We fix an ordinal α with $|\alpha| > \kappa$, and for each $p \in P$ we let $\mathcal{L}_{\alpha,p}$ denote the well-monotonic graph over $L_{\alpha,p} = \alpha$ introduced in the first section for co-Büchi objectives translated to C_p, formally

$$\lambda_p \xrightarrow{c} \lambda'_p \in \mathcal{L}_{\alpha,p} \quad \iff \quad \begin{array}{lll} c = 2p+1 & \text{and} & \lambda_p > \lambda'_p \qquad \text{or} \\ c = 2p & \text{and} & \lambda_p \geqslant \lambda'_p. \end{array}$$

Then the lexicographical product $\mathcal{L}_\alpha$ of the $\mathcal{L}_{\alpha,p}$'s is the graph over $L_\alpha = \alpha^h$ given by

$$\begin{array}{lll} \lambda \xrightarrow{2p} \lambda' \text{ in } \mathcal{L}_\alpha & \iff & \lambda \geqslant_p \lambda' \qquad \text{and} \\ \lambda \xrightarrow{2p+1} \lambda' \text{ in } \mathcal{L}_\alpha & \iff & \lambda >_p \lambda', \end{array}$$

which coincides with Walukiewicz's [Wal96] well-known notion of *signatures assignments*. Combining Lemma 2.8 and Theorem 2.3, $\mathcal{L}_\alpha$ is $\text{Parity}_{[1,2h]}$-universal for the class of all graphs of cardinality $< |\alpha|$. This proves via Theorem 1.1 that arbitrary parity games are positionally determined, and the proof essentially coincides with that of [EJ91]. A more direct proof is given in [Kop06], which is inductive on h as is ours.

An extension of the signature-based proof priorities in ω with the min-parity condition was presented by Grädel and Walukiewicz [GW06], over countable vertex-coloured arenas. Their paper focuses on bi-positionality, and the counter-example they provide for edge-coloured arenas applies only to the player who is declared a winner when no priority is seen infinitely often; it is thus not ruled out that the opponent has positional strategies. In this vein it would be very interesting to understand whether Theorem 2.3 can be extended from finite to ordinal lexicographical product, with an adequate definition, but we leave this to future work. Such an extension is also suggested by general results of Büchi [Büc83].

Here, the choice of co-Büchi rather than Büchi is completely arbitrary, and the graph obtained by lexicographical combination of the construction given in Section 1 for Büchi conditions yields a $\text{Parity}_{[0,2h-1]}$-universal graph which coincides with the one above when the colours are restricted to $[1, 2h-1]$.

This also shows that lexicographical products of montonic graphs which are universal over graphs of finite or bounded degree do not have this property, otherwise one could obtain a countable graph which is Co-Büchi-universal for graphs of finite or bounded degree, contradicting Lemma 2.10. In the proof of Theorem 2.3 below, closing G to obtain G_0 may produce a graph of large degree even if G is not.

Lexicographical mean-payoff games. We now quickly discuss lexicographical products of threshold mean-payoff games. Fix n and N in ω. For $p \in P = \{1, \dots, h\}$ we let C_p be a copy of $[-N, N]$, whose elements we denote by t_p for $t \in [-N, N]$, and we let C be the disjoint union of the C_p's. We let W_p denote the (prefix-independent) threshold mean-payoff objective $\mathrm{MP}^{\leqslant 0}$ over C_p and $\mathcal{L}_p$ be a copy of the finite monotonic graph $\mathcal{L}_{[0,(n-1)N]}$ from Corollary 2.3 for each p, whose elements we denote by ℓ_p for $\ell \in \omega$,

$$\ell_p \xrightarrow{t_p} \ell'_p \text{ in } \mathcal{L}_p \qquad \Longleftrightarrow \qquad t_p \leqslant \ell_p - \ell'_p.$$

Then the lexicographical product $\mathcal{L}$ of the $\mathcal{L}_p$'s is defined over $[0, (n-1)N]^h$ by

$$\ell \xrightarrow{t_p} \ell' \text{ in } \mathcal{L} \qquad \Longleftrightarrow \qquad \ell >_{p+1} \ell' \text{ or } (\ell =_{p+1} \ell' \text{ and } t_p \leqslant \ell_p - \ell'_p).$$

The lexicographical product W of the W_p's is interpreted as

$$W = \left\{w \in C^\omega \mid \mathrm{MP}(w_p) \leqslant 0 \text{ where } p = \max\{p \in P \mid w_p \text{ is infinite}\}\right\}.$$

In words, Eve should ensure that the dimension which corresponds to the largest index p that has infinitely many occurrences, has non-positive mean-payoff (profiles corresponding to indices with lower priority are allowed to diverge arbitrarily).

Combined with Corollary 2.3 the theorem yields that the finite well-monotonic graph $\mathcal{L}$ is W-universal for the class of all $[-N, N]$-graphs of size $\leqslant n$. This gives a value iteration algorithm (see Chapter 4) with runtime $O(m(nN)^h)$ for lexicographic mean-payoff games. Several different formalisms have been considered for lexicographic mean-payoff games [BCH+09; BMR14; CJL+17]. Ours is similar to the one in the third paper (which also mentions an unpublished related work of Colcombet and Niwiński on lexicographic energy games), and displays the same complexity; we do not know to what extent the two notions are interreducible.

4.3 Proof of Theorem 2.3

We fix a cardinal κ, and a familly of well-monotonic graphs $\mathcal{L}_p$ which are W_p-universal for the class of all C_p-graphs of cardinality $\leqslant \kappa$. Recall that the W_p's are assumed to be prefix-independent, and so is W. There are two things to show: that $\mathcal{L}$ satisfies W and that $\mathcal{L}$ embeds all graphs of cardinality $\leqslant \kappa$ which satisfy W. We start with the first property.

Lemma 2.17

It holds that $\mathcal{L}$ satisfies W.

The fact that the $\mathcal{L}_p$'s are well-ordered is crucial here.

Proof. Consider an infinite path $\pi : \ell^0 \xrightarrow{c^0} \ell^1 \xrightarrow{c^1}$ in $\mathcal{L}$, let $w = c^0 c^1 \dots$ be its colouration, for all i let $p^i \in P$ denote the priority of c^i and let p_0 denote the maximal priority that appears infinitely often. We aim to prove that w_{p_0} belongs to W_{p_0}. We decompose π as

$$\pi : \ell^0 \overset{w^0}{\rightsquigarrow} \ell^{i_0} \xrightarrow{c^{i_0}} \ell^{i_0+1} \overset{w^1}{\rightsquigarrow} \ell^{i_1} \xrightarrow{c^{i_1}} \ell^{i_1+1} \overset{w^2}{\rightsquigarrow} \dots,$$

for all j which ranges over ω, c^{i_j} has priority p_0 and for all $j \geqslant 1$, w^j has only colours of priority $< p_0$.

In particular, for all $i \geqslant i_0$ it holds that c^i has priority $\leqslant p_0$, and therefore $\ell^i \geqslant_{p_0+1} \ell^{i+1}$. Recall that $\geqslant_{p_0+1}$ defines a well-order over $L_{\geqslant p_0+1}$ thus $\ell^i_{\geqslant p_0+1}$ is ultimately constant, and we let j_0 be such that

$$\forall i \geqslant i_{j_0}, \qquad\qquad \ell^i =_{p_0+1} \ell^{i_{j_0}}.$$

Now for all $j \geqslant j_0$, and for all $i \in [i_j + 1, i_{j+1}]$, c^i is of priority $< p_0$ thus $\ell^i \geqslant_{p_0} \ell^{i+1}$, and since moreover $\ell^i =_{p_0+1} \ell^{i+1}$ it holds that $\ell^i_{p_0} \geqslant \ell^{i+1}_{p_0}$. Hence by transitivity, $\ell^{i_j+1}_{p_0} \geqslant \ell^{i_{j+1}}_{p_0}$.

Moreover, again for $j \geqslant j_0$ and since c^{i_j} has priority p_0 and $\ell^{i_j} =_{p_0+1} \ell^{i_j+1}$, we have $\ell^{i_j}_{p_0} \xrightarrow{c^{i_j}} \ell^{i_j+1}_{p_0}$ in $\mathcal{L}_{p_0}$. Therefore

$$\ell^{i_{j_0}}_{p_0} \xrightarrow{c^{i_{j_0}}} \ell^{i_{j_0}+1}_{p_0} \geqslant \ell^{i_{j_0+1}}_{p_0} \xrightarrow{c^{i_{j_0+1}}} \ell^{i_{j_0+1}+1}_{p_0} \geqslant \ell^{i_{j_0+2}}_{p_0} \xrightarrow{c^{i_{j_0+2}}} \dots$$

holds in $\mathcal{L}_{p_0}$, and thus by monotonicity of $\mathcal{L}_{p_0}$,

$$\ell^{i_{j_0}}_{p_0} \xrightarrow{c^{i_{j_0}}} \ell^{i_{j_0+1}}_{p_0} \xrightarrow{c^{i_{j_0+1}}} \ell^{i_{j_0+2}}_{p_0} \xrightarrow{c^{i_{j_0+2}}} \dots$$

defines a path in $\mathcal{L}_{p_0}$. Hence $(w_{p_0})_{\geqslant i_{j_0}} = c^{i_{j_0}} c^{i_{j_0+1}} \dots \in W_{p_0}$ since $\mathcal{L}_{p_0}$ satisfies W_{p_0} and therefore $w_{p_0} \in W_{p_0}$ since it is prefix-decreasing, and we conclude that $w \in W$. □

We now show the second property, namely that under the assumption of the theorem, $\mathcal{L}$ embeds all graphs of cardinality $\leqslant \kappa$ which satisfy W. For clarity, we break the proof into a few steps.

Given $p_0 \in P$ we let $C_{\leqslant p_0} = \cup_{p=1}^{p_0} C_p$, $\mathcal{L}_{\leqslant p_0} = \otimes_{p=1}^{p_0} \mathcal{L}_p$ and $W_{\leqslant p_0} = \otimes_{p=1}^{p_0} W_p$. Note that for $p_0 \geqslant 2$ we have $\mathcal{L}_{\leqslant p_0} = \mathcal{L}_{\leqslant p_0 - 1} \otimes \mathcal{L}_{p_0}$ and likewise for $W_{\leqslant p_0}$.

We prove by induction on $p_0 \in P$ that $\mathcal{L}_{\leqslant p_0}$ embeds all $C_{\leqslant p_0}$-graphs of cardinality $\leqslant \kappa$ which satisfy W. This is clear for $p_0 = 1$ since we have $\mathcal{L}_{\leqslant p_0} = \mathcal{L}_1$ which is $W_{\leqslant 1} = W_1$-universal for the class of graphs of cardinality $\leqslant \kappa$. We now let $p_0 \geqslant 2$, assume the result known for $p_0 - 1$ and let G denote a $C_{\leqslant p_0}$-graph over V of cardinality $\leqslant \kappa$ which satisfies $W_{\leqslant p_0}$. The proof is illustrated in Figure 2.9.

We let G_0 denote the C_{p_0}-pregraph over V given by for all $v, v' \in V$ and $c_0 \in C_{p_0}$,

$$v \xrightarrow{c_0} v' \text{ in } G_0 \qquad \iff \qquad \exists v_1, v_2 \in V, w_1, w_2 \in C^*_{\leqslant p_0 - 1}, \quad v \overset{w_1}{\rightsquigarrow} v_1 \xrightarrow{c_0} v_2 \overset{w_2}{\rightsquigarrow} v' \text{ in } G.$$

Intuitively, we have closed in G important edges (those of maximal priority p_0) on both sides under paths comprised of less important edges, and then restricted to important edges.

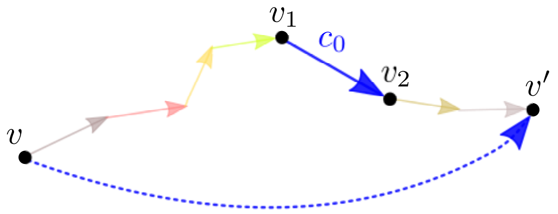


Figure 2.8: Illustrating the definition of G_0: a c_0-edge connects v to v' in G_0 if and only if there is a path in G from v to v' containing a c_0-edge.

Note that G_0 is not a graph in general: vertices which do not have a path visiting an edge of priority p_0 in G are sinks in G_0. This is not an issue thanks to prefix-decreasingness of W_{p_0} (see Chapter 1).

Lemma 2.18

The pregraph G_0 satisfies W_{p_0}.

Proof. Consider an infinite path in G_0. It is of the form

$$\pi_0 : v_0 \xrightarrow{c_1} v_3 \xrightarrow{c_4} v_6 \xrightarrow{c_7} \dots$$

where $c_1, c_4, c_7, \dots \in C_{p_0}$, and there exist $v_1, v_2, v_4, v_5, v_7, v_8, \dots \in V$ and $w_0, w_2, w_3, w_5, w_6, w_8, \dots \in C^*_{\leqslant p_0 - 1}$ such that

$$\pi : v_0 \overset{w_0}{\rightsquigarrow} v_1 \xrightarrow{c_1} v_2 \overset{w_2}{\rightsquigarrow} v_3 \xrightarrow{w_3} v_4 \xrightarrow{c_4} v_5 \overset{w_5}{\rightsquigarrow} v_6 \overset{w_6}{\rightsquigarrow} v_7 \xrightarrow{c_7} v_8 \overset{w_8}{\rightsquigarrow} \dots$$

defines a path in G. Therefore π satisfies W, and since p_0 is the maximal priority appearing infinitely often on π, this means that $w_{p_0} \in W_{p_0}$, where $w = w_0 w_1 \dots$ is the colouration of π. This yields the wanted result since w_{p_0} is the colouration of π_0. □

We let $\psi_0 : G_0 \to L_{p_0}$ denote the evaluation of G_0 which is well defined thanks to the lemma. Now comes the crucial claim.

Lemma 2.19 (No small edge goes up in ψ_0)

If $v, v' \in V$ are such that $\psi_0(v) < \psi_0(v')$ then there is no edge $v \xrightarrow{c} v'$ in G with priority $< p_0$.

Proof. Assume for contradiction that there is such an edge. Then in G_0 for all $c_0 \in C_{p_0}$, any c_0-successor of v' is also a c_0-successor of v. Therefore,

$$\psi_0(v) = \mathrm{Upd}^{\mathcal{L}_{p_0}}_{G_0}(\psi_0)(v) = \sup_{v \xrightarrow{c_0} u' \text{ in } G_0} \rho(\psi_0(u'), c_0) \geqslant \sup_{v' \xrightarrow{c_0} u' \text{ in } G_0} \rho(\psi_0(u'), c_0) = \psi_0(v'),$$

a contradiction. □

Now for each $\ell_{p_0} \in L_{p_0}$, we let $G^{\ell_{p_0}}$ denote the restriction of G to $V^{\ell_{p_0}} = \psi^{-1}(\ell_{p_0})$ and to edges of priority $< p_0$. Again, $G^{\ell_{p_0}}$ is only a pregraph in general, which is not an issue since $W_{\leqslant p_0 - 1}$ is prefix-decreasing. Also it may be empty for some ℓ_{p_0}'s which is not an issue either.

For each $\ell_{p_0} \in L_{p_0}$, the graph $G^{\ell_{p_0}}$ satisfies $W_{\leqslant p_0}$ since G does, and it even satisfies $W_{\leqslant p_0 - 1}$ since it has only edges of priority $< p_0$ and $W_{\leqslant p_0} \cap C_{\leqslant p_0 - 1} = W_{\leqslant p_0 - 1}$. Therefore, by our induction hypothesis, there exists for each $\ell_{p_0} \in \mathcal{L}_{p_0}$ a morphism $\phi^{\ell_{p_0}}$ from $G^{\ell_{p_0}}$ to $\mathcal{L}_{\leqslant p_0 - 1}$.

We now define a map $\phi : V \to L_{\leqslant p_0}$ by

$$\phi(v)_{\leqslant p_0 - 1} = \phi^{\psi_0(v)}(v) \qquad \text{and} \qquad \phi(v)_{p_0} = \psi_0(v).$$

The following result concludes the proof of Theorem 2.3.

Lemma 2.20

The map ϕ defines a morphism of G in $\mathcal{L}_{\leqslant p_0}$.

Proof. We first recall that by definition of $\mathcal{L}_{\leqslant p_0} = \mathcal{L}_{\leqslant p_0 - 1} \otimes \mathcal{L}_{p_0}$, we have for $c_0 \in C_{p_0}$,

$$\ell \xrightarrow{c_0} \ell' \text{ in } \mathcal{L}_{\leqslant p_0} \qquad \iff \qquad \ell_{p_0} \xrightarrow{c_0} \ell'_{p_0} \text{ in } L_{p_0},$$

and for $c \in C_{\leqslant p_0 - 1}$,

$$\ell \xrightarrow{c} \ell' \text{ in } \mathcal{L}_{\leqslant p_0} \qquad \iff \qquad \ell_{p_0} > \ell'_{p_0} \text{ or } (\ell_{p_0} = \ell'_{p_0} \text{ and } \ell_{\leqslant p_0 - 1} \xrightarrow{c} \ell'_{\leqslant p_0 - 1} \text{ in } \mathcal{L}_{\leqslant p_0 - 1}).$$

We have to verify that

$$v \xrightarrow{c} v' \text{ in } G \qquad \implies \qquad \phi(v) \xrightarrow{c} \phi(v') \text{ in } \mathcal{L},$$

and we separate two cases.

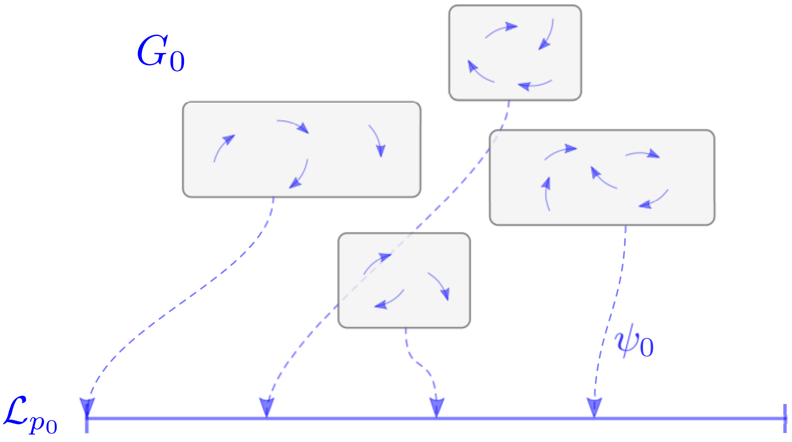


Figure 2.9: An illustration for the proof. The graph G is first turned into a C_{p_0}-graph G_0 by closing by transitivity with other edges. Then G_{p_0} is mapped into L_{p_0} by universality. This defines components in G, which are treated separately by induction.

- If c has priority p_0 then $v \xrightarrow{c} v'$ in G_0 thus $\psi_0(v) \xrightarrow{c} \psi_0(v')$ in $\mathcal{L}_{p_0}$ which yields the result.
- Otherwise we know by Lemma 2.19 that $\psi_0(v) \geqslant \psi_0(v')$. If this inequality is strict then the definition of $\mathcal{L}_{\leqslant p_0}$ (recalled above) gives the result. Otherwise the fact that $\phi^{\psi_0(v)}$ is a morphism from $G^{\psi_0(v)}$ to $L_{\leqslant p_0 - 1}$ concludes. □

Structuration results 3

We have seen in Chapter 1 that if a valuation admits well-monotonic universal graphs, then it is positional. We now study converse statements: can we guarantee that a given positional valuation admits well-monotonic universal graphs?

Existence of arbitrary universal graphs (even colouration-universal) for a given class of graphs (of bounded cardinality) is straightforward: it suffices to consider a disjoint union of all graphs from the class, up to isomorphism. Our main results in this chapter state that if val is positionally determined over large enough arenas, then one can turn any graph into a well-monotonic one by adding sufficiently many edges and quotienting, while preserving val. This establishes the wanted converse.

Section 1 introduces colour neutrality, states our two structuration results, discusses their consequences and gives an overview of the proofs, which are broken in two steps. Section 2 deals with the second step which is easier and common to both proofs. Finally, Section 3 gives the core of the proofs, which rely on two different variants of *choice arenas*, one for each result.

1 Statement of the results and discussion

Before stating our structuration results we need to introduce neutral colours.

Colour neutrality. Given two infinite words $w, w' \in C^\omega$, we say that w' is *obtained from w by adding c's* if there exist a sequence of natural numbers $n_0, n_1 \dots \in \mathbb{N}$ such that

$$w' = w_0 c^{n_0} w_1 c^{n_1} w_2 c^{n_2} \dots .$$

For example, both 0111010^ω and $(01)^\omega$ are obtained from 0^ω by adding 1's, but 01^ω is not. We say that w' is obtained from w by *weakly* adding c's if the sequence $n_0 n_1 \dots$ is bounded. For convenience, we also say in this case that w is (weakly) obtained from w' by *removing* c's.

Fix a C-valuation val and let $c \in C$. We say that c is *good for Eve* (with respect to val) if whenever w' is obtained from w by adding c's, it holds that $\mathrm{val}(w') \leqslant \mathrm{val}(w)$. In words, adding occurrences of c's does not increase the valuation. We define being *good for Adam* symmetrically. We say that c is *neutral* if it is good for both players: if w' is obtained from w by adding c's then $\mathrm{val}(w') = \mathrm{val}(w)$. These notions have weak counterparts, for instance c is weakly good for Adam if weakly adding c's does not decrease the valuation. Note that being neutral says nothing about the value of words of the form uc^ω.

We say that a colour c is *ultimately good for Eve* (with respect to val) if for all finite words $u \in C^*$ it holds that

$$\mathrm{val}(uc^\omega) = \inf_{v \in C^\omega} \mathrm{val}(uv),$$

and that it is *strongly neutral* if it is neutral and ultimately good for Eve. Note that weakly neutral, neutral, and strongly neutral, are three different stronger and stronger notions.

We discuss a few examples.

- For a parity condition, even priorities are good for Eve and odd priorities are good for Adam. The smallest priority is neutral, and it is ultimately good for Eve (and therefore strongly neutral) if and only if it is even.

- For the energy valuation, non-positive weights are good for Eve and non-negative weights are good for Adam. Therefore 0 is the only neutral colour, and 0 is ultimately good for Eve, hence strongly neutral.

- For the threshold mean-payoff objective $\mathrm{MP}^{\leqslant 0}$, non-positive weights are good for Eve and positive weights are good for Adam. But 0 is not neutral since $101001000\cdots \in \mathrm{MP}^{\leqslant 0}$ is obtained from $111\cdots \notin \mathrm{MP}^{\leqslant 0}$ by adding 0's therefore 0 is not good for Adam. However it is weakly good for Adam and thus weakly neutral (it is also ultimately good for Eve but this has no importance in this case).

- For an arbitrary lexicographical product W of $W_1 \subseteq C_1$ and $W_2 \subseteq C_2$, all properties are preserved from W_1 to W for colours in C_1. This is not always the case for colours in C_2, however it does hold that if $c_2 \in C_2$ is (weakly) good for Eve and ultimately good for Eve for W_2 then the same holds with respect to W.

Given a valuation val over C and a fresh letter $0 \notin C$, there is a unique extension val$'$ of val over $C' = C \cup \{0\}$ for which 0 is strongly neutral, defined in the obvious way. It is not known in general whether val$'$ is positionally determined for Eve when val is, even for prefix-independent objectives W (see [Kop06] and also [Cas21] for further discussion).

Stated differently, we do not know whether the existence of a (weakly) neutral letter imposes a meaningful restriction on the class of objectives which are positionally determined or not. We know however by the previous chapter that if W is prefix-independent and has well-monotonic universal graphs over given classes of graphs, then so does W', by lexicographical product with the trivial objective. Our results motivate further investigation for this question.

Structuration results and consequences. Given a graph G over V and a valuation val, a val-*structuration* of G is a well-monotonic graph G' over V' such that $|V'| \leqslant |V|$ and G has a val-preserving morphism into G'.

Our first structuration result is easier to prove but fails for infinite graphs.

Theorem 3.1 (Finite structuration via saturation)

If val has a weakly neutral colour and is uniformly positionally determined over finite arenas then any finite graphs admits a val-structuration.

The second result drops the finiteness hypothesis, at the price of strengthening the hypotheses on the neutral colour and the positionality requirement.

Theorem 3.2 (Strong structuration via multiple choice arenas)

Fix a (possibly infinite) graph G and assume that val has a strongly neutral colour and that it is uniformly positionally determined over arenas of cardinality $|V| + 2^{|V|}$ and degree $\max(|V|, \deg(G))$. Then G has a val-structuration.

We now give two (structural) corollaries of Theorem 3.2, (algorithmic) consequences of Theorem 3.1 will appear in the next chapter. We first state our main contribution.

Corollary 3.1 (Existence of universal well-monotonic graphs characterise positionality)

Let val be a valuation with a strongly neutral colour. Then val is uniformly positionally determined over all arenas if and only if for any cardinal κ there exists a well-monotonic graph which is val-universal for all graphs of cardinality $\leqslant \kappa$.

Proof. The converse implication follows from Theorem 1.1. Let κ be a cardinal and let G be the disjoint union of all C-graphs of cardinal κ up to isomorphism. By Theorem 3.2 G has a val-preserving morphism into a well-monotonic graph $\mathcal{L}$. Now $\mathcal{L}$ is val-universal for all graphs of cardinal κ since G is, by composition of val-preserving morphisms. □

Our second corollary provides a nice closure property, which we like to see as a proof of concept for our approach.

Corollary 3.2 (Closure under lexicographical product)

The class of objectives which are positionally determined over all arenas, prefix-independent, and have a strongly neutral colour is closed under lexicographical product.

As far as we know this result is novel, and we do not know if it admits a more direct proof.

Proof. Let W_1 and W_2 be such objectives. By Corollary 3.1 they admit well-monotonic graphs $\mathcal{L}_1^\kappa$ and $\mathcal{L}_2^\kappa$ which are universal with respect to their respective objectives for graphs of cardinality $\leqslant \kappa$. By Theorem 2.3 $\mathcal{L}_1 \otimes \mathcal{L}_2$ is $W_1 \otimes W_2$-universal for graphs of cardinality $\leqslant \kappa$, and therefore W is positionally determined. It is easy to see that the strongly neutral colour $0_1 \in C_1$ for W_1 is also strongly neutral for W. □

Overview of the proofs. Both constructions we propose are done in two steps. We fix a colour c (which will be chosen to be neutral) and

(i) add *many* c-edges to G while preserving val; then

(ii) add even more edges by closing around c-edges (this will be made precise below), and quotient by $\xrightarrow{c}$-equivalence.

The second step is generic: such a closure does not increase val as long as c is good for Adam. For it to produce a well-structured graph however, we need to guarantee that there are already sufficiently many c-edges, which were added in the first step (see Lemma 3.2 for formal statement).

The first step is harder and differs for both constructions; we need ways of introducing many c-edges. In the case where G is finite, one may simply add c-edges arbitrarily as long as val is preserved. This process terminates and produces a *c-saturation* G' of G, which one can show has good properties (many c-edges) when val is positional over finite arenas. The crucial fact that c-saturated graphs have many c-edges, which relies on the fact that c is (weakly) good for Eve, is shown using single choice arenas, and it can be generalised to infinite graphs (see Theorem 3.3 for details).

However, existence of a c-saturation of G is no longer guaranteed if G is infinite which is why we need another means of introducing many c-edges to obtain Theorem 3.2. For this we use *multiple choice arenas*, which generalise single choice arenas but require a strongly neutral colour. Multiple choice arenas appear to be a robust tool to exploit positionality for Eve, and we believe that they may be of independent interest besides the proof of Theorem 3.2.

Section 2 discusses the second step and introduces saturation. In Section 3 we present single and multiple choice arenas, and exploit them to prove the two theorems. From now on, we fix a valuation $\mathrm{val} : C^\omega \to X$.

2 Closure and saturation

We start by giving details for the second step which is common to both proofs.

c-closures. Given a C-graph G and a colour $c \in C$ we define the *c-closure* G' of G to be the C-graph over V given by

$$v \xrightarrow{c'} v' \text{ in } G' \quad \iff \quad \exists n_1, n_2, \in \mathbb{N}, \exists v_1, v_2 \in V, \quad v \overset{c^{n_1}}{\rightsquigarrow} v_1 \xrightarrow{c'} v_2 \overset{c^{n_2}}{\rightsquigarrow} v'.$$

In words, G' is obtained from G by closing $\xrightarrow{c}$ by transitivity and with other edges on both sides. Note that if G has finite size n then n_1 and n_2 can equivalently be chosen bounded by n.

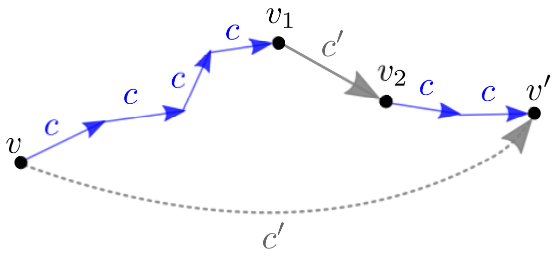


Figure 3.1: An illustration of the c-closure G' of G. Note that the situation is similar, but different, from that of Figure 2.8: here, G' is a C-graph (in particular, no edge is removed), and moreover the closure is done with respect to only one colour, namely c.

Lemma 3.1 (Preservation of val)

If c is good for Adam then the identity morphism from G to its c-closure G' is val-preserving. The same result holds for finite G if c is only weakly good for Adam.

Proof. It is clear that the identity is a morphism since G' is obtained from G by adding edges. Consider a path

$$\pi' : v_0 \xrightarrow{c'_1} v_3 \xrightarrow{c'_4} v_6 \xrightarrow{c'_7} \dots$$

in G'. By definition there exist $v_1, v_2, v_4, v_5, v_7, v_8, \dots$ in V and $n_0, n_2, n_3, n_5, n_6, n_8, \dots \in \mathbb{N}$ such that

$$\pi : v_0 \overset{c^{n_0}}{\rightsquigarrow} v_1 \xrightarrow{c'_1} v_2 \overset{c^{n_2}}{\rightsquigarrow} v_3 \overset{c^{n_3}}{\rightsquigarrow} v_4 \xrightarrow{c'_4} v_5 \overset{c^{n_5}}{\rightsquigarrow} v_6 \overset{c^{n_6}}{\rightsquigarrow} v_7 \xrightarrow{c'_7} v_8 \overset{c'^{n_8}}{\rightsquigarrow} \dots$$

is a path in G. Now observe that $\mathrm{col}(\pi')$ is obtained from $\mathrm{col}(\pi)$ by removing c's, and thus $\mathrm{val}(\pi') \leqslant \mathrm{val}(\pi)$. For the second statement, it suffices to choose the n_i's smaller than the size of G. □

Sufficiently many edges. We say that G has *sufficiently many c-edges* if it holds that

$$\forall S \subseteq V, \big[S \neq \varnothing \implies \exists s \in S, \forall s' \in S\backslash\{s\}, \quad s' \xrightarrow{c} s \text{ in } G\big].$$

Stated differently the relation "having a c-edge" over V is well-founded.

Lemma 3.2 (Sufficiently many edges implies structuration)

If G has sufficiently many c-edges then its c-closure G' has a structuration.

Proof. We study the relation $\geqslant$ induced over V by the reflexive closure of $\xrightarrow{c}$ in G', formally

$$v \geqslant v' \qquad \iff \qquad \big[v = v' \text{ or } v \xrightarrow{c} v' \text{ in } G'\big].$$

Let $v, v', v'' \in V$ and $c' \in C$ be such that $v \geqslant v' \xrightarrow{c'} v''$ in G'. If $v = v'$ then $v \xrightarrow{c'} v''$ in G'. Otherwise there exist $v_1, v_2, v_3, v_4 \in V$ and $n_1, n_2, n_3, n_4 \in \mathbb{N}$ such that

$$v \overset{c^{n_1}}{\rightsquigarrow} v_1 \xrightarrow{c} v_2 \overset{c^{n_2}}{\rightsquigarrow} v' \overset{c^{n_3}}{\rightsquigarrow} v_3 \xrightarrow{c'} v_4 \overset{c^{n_4}}{\rightsquigarrow} v'' \text{ in } G,$$

thus

$$v \overset{c^{n_1+1+n_2+n_3}}{\rightsquigarrow} v_3 \xrightarrow{c'} v_4 \overset{c^{n_4}}{\rightsquigarrow} v'' \text{ in } G$$

and therefore $v \xrightarrow{c'} v''$ in G'. This proves left composition in G', and the proof of right composition is omitted since it follows exactly the same lines. As the particular case where $c = c'$ this also shows transitivity of $\geqslant$, which makes it a preorder. It is even total since G (and therefore G') has sufficiently-many c-edges: pairs have a minimal element.

We let $\sim$ denote the associated equivalence relation over V, which is given by $v \sim v'$ if and only if $v = v'$ or $v \xrightarrow{c} v' \xrightarrow{c} v$ in G'. Observe that by left composition in G' equivalent vertices have the same successors, and by right composition they have the same predecessors. Therefore the graph G'' over $V/\sim$ given by

$$[v]_\sim \xrightarrow{c'} [v']_\sim \text{ in } G'' \qquad \iff \qquad v \xrightarrow{c'} v' \text{ in } G$$

is well defined, and it is monotonic with respect to $\geqslant$. Well-orderedness of $\geqslant$ over $V/\sim$ holds by definition since G (and thus G') has sufficiently many c-edges. The map $v \mapsto [v']_\sim$ defines a morphism from G' to G'' which is colouration-preserving by definition. □

Combining the two previous lemmas gives the following result.

Corollary 3.3 (Second step)

If c is good for Adam and G has sufficiently many c-edges then it has a structuration. This is also true if G is finite and c is only weakly good for Adam.

Saturation. Given a graph G over V and an edge $e \in V \times C \times V$ we let G_e denote the game obtained by adding e to G (if e belongs to G then $G_e = G$). We say that a graph G is *c-saturated* with respect to val if it holds that for all c-edges which do not belong to G there exists a vertex v such that

$$\mathrm{val}_{G_e}(v) > \mathrm{val}_G(v).$$

In words, the addition of any new c-edge to G entails an increase in val.

Lemma 3.3 (c-saturation of a finite graphs)

For all finite graphs G over V, there exists a c-saturated graph G' over V such that G has a val-preserving morphism into G'.

Informally, we simply add arbitrary c-edges to G until it becomes saturated.

Proof. Let $E \subseteq V \times C \times V$ denote the set of edges of G. The set of all graphs obtained by adding c-edges to G and which have the same val-values as G is finite since V is. Note that for all $e \notin E$ and for all $v \in V$ it holds that $\mathrm{val}_{G_e}(v) \geqslant \mathrm{val}_G(v)$, or stated differently val is monotonous with respect to the operation of adding edges. Therefore a maximal element G' of the above set exists by finiteness and is c-saturated by definition. □

Note that this proof fails for infinite graphs; we do not know if they admit saturations in general, even with further assumption on val (see Figure 3.2).

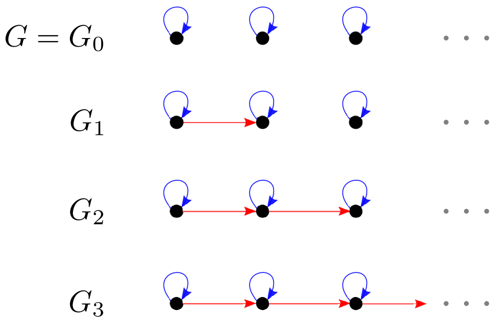


Figure 3.2: Illustrating the limits of saturation with the Co-Büchi condition over safe (blue) and bad (red). We start with an infinite graph G satisfying Co-Büchi, and iteratively add bad-edges leading to an increasing chain $G_0, G_1, G_2 \ldots$ of graphs satisfying Co-Büchi. Any upper bound to this chain contains an infinite bad path, and therefore does not satisfy Co-Büchi. Stated differently, it is not clear how to saturate infinite graphs in a val-preserving way in general.

We may now state our main result about saturation.

Theorem 3.3 (Saturating adds sufficiently many edges)

Let $c \in C$ be weakly good for Eve (with respect to val) and consider a graph G over V which is c-saturated (with respect to val). If val is positionally determined for Eve over arenas of size $|V| + 1$ and degree $\max(|V|, \deg(G))$ then G has sufficiently-many c-edges.

Together with Lemma 3.3 and Corollary 3.3 this implies the finite structuration result in Theorem 3.1.

Another way to interpret the theorem is that in general, given a colour c which is weakly good for Eve (and under the right positionality assumption), and given two vertices v, v' in a graph, one can always add a c-edge either from v to v' or the other way, while preserving val. This even generalises to (potentially infinite) subsets S in infinite graphs: they all have an element towards which c-edges can be added from all $s \in S$ (except the chosen element, unless c is also ultimately good for Eve in which case c-loops can safely be added).

We now introduce choice arenas for proving Theorems 3.3 and 3.2.

3 Choice arenas

Fix a C-graph G over V which we see as a val-game fully controlled by Adam. We will construct a game which is not harder for Eve with respect to val, but where the natural strategy guaranteeing this fact requires memory. This will allow to use uniform positional determinacy of val as a lever: an optimal positional strategy yields a set of edges which can be added to G without increasing val.

We introduce two similar yet incomparable variants.

- *Single choice arenas* require a colour which is weakly good for Eve, they have cardinality $|V|+1$ and a single Eve vertex. They are used to prove Theorem 3.3.
- *Multiple choice arenas* require a colour which is good for Eve and ultimately good for Eve, and they have cardinality $|V| + 2^{|V|}$ among which $2^{|V|}$ Eve vertices. They are used to prove Theorem 3.2.

3.1 Single choice arenas

Fix a non-empty subset S of vertices of G, and a colour $c \in C$ which is weakly good for Eve with respect to val. The *single choice arena* given by G, S and c is the arena $G_{\text{sch}(S)}$ over $V \cup \{S\}$ with $V_{\text{Adam}} = V$ and $V_{\text{Eve}} = \{S\}$ given by

$$\begin{aligned}
v \xrightarrow{c'} v' \text{ in } G_{\text{sch}(S)} \quad &\iff \quad v \xrightarrow{c'} v' \text{ in } G\\
v \xrightarrow{c'} S \text{ in } G_{\text{sch}(S)} \quad &\iff \quad \exists s \in S, v \xrightarrow{c'} s \text{ in } G\\
S \xrightarrow{c'} v \text{ in } G_{\text{sch}(S)} \quad &\iff \quad c' = c \text{ and } v \in S.
\end{aligned}$$

It is illustrated in Figure 3.3.

Intuitively Adam follows a path in G with the additional possibility from any c'-predecessor v of a vertex v' of S to instead go to S, seeing colour c'. It is then left to Eve to choose a successor from S, seeing colour c. A natural choice is v' which guarantees a smaller val-value by our assumption on c.

Lemma 3.4 (Single choice arenas are easy)

For all $v \in V$, it holds that $\text{val}_{G_{\text{sch}(S)}}(v) \leqslant \text{val}_G(v)$.

The converse inequality is (not important and) obvious : any path in G' can be realised by Adam in $G_{\text{sch}(S)}$, simply by never visiting $V_{\text{Eve}} = \{S\}$.

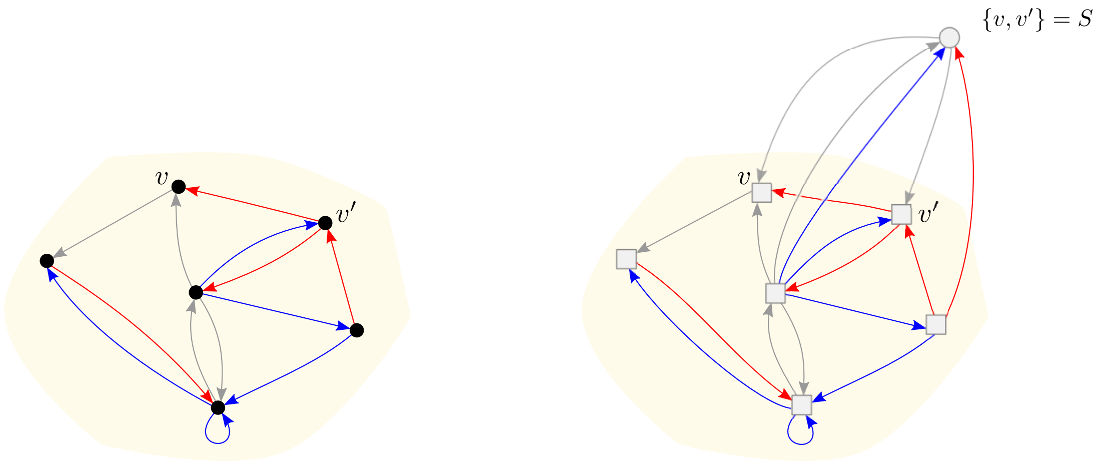


Figure 3.3: On the left, a $\{$blue, red, gray$\}$-arena G. On the right, the single choice arena given by $c =$ gray and $S = \{v, v'\}$.

Proof. Let $v_0 \in V$. Consider an arbitrary Eve-strategy σ from v_0 satisfying

$$\sigma(\pi\ (v, c', S)) = (S, c, v'),$$

where $v \xrightarrow{c'} v'$ in G, for all finite paths π. In words, when arriving in S from v with colour c', Eve picks a c'-successor v' of v in G.

Let π' be a path in $G_{\text{sch}(S)}$ consistent with σ. It is of the form

$$\pi' : v_0 \overset{w_0}{\rightsquigarrow} v_1 \xrightarrow{c_1'} S \xrightarrow{c} v_2 \overset{w_2}{\rightsquigarrow} v_3 \xrightarrow{c_3'} S \xrightarrow{c} v_4 \overset{w_4}{\rightsquigarrow} \dots$$

for some (possibly empty, finite, or infinite) sequence of vertices $v_1, v_2, v_3, v_4, \dots$ satisfying for all i even

$$v_i \overset{w_i}{\rightsquigarrow} v_{i+1} \text{ in } G, \qquad v_{i+1} \xrightarrow{c_{i+1}'} v_{i+2} \text{ in } G \quad \text{ and } \quad v_{i+2} \in S$$

Moreover if the sequence is finite, its last index ℓ is even and such that $v_\ell \overset{w_\ell}{\rightsquigarrow}$ in G.

Thus the path obtained by-passing the occurrences of S, formally

$$\pi : v_0 \overset{w_0}{\rightsquigarrow} v_1 \xrightarrow{c_1'} v_2 \overset{w_2}{\rightsquigarrow} v_3 \xrightarrow{c_3'} v_4 \overset{w_4}{\rightsquigarrow} \dots$$

defines an infinite path in G which moreover satisfies that $w' = \text{col}(\pi')$ is obtained from $w = \text{col}(\pi)$ by weakly adding c's. Hence we have $\text{val}(w') \leqslant \text{val}(w)$ since c is good for Eve and thus

$$\text{val}_{G_{\text{sch}(S)}}(v_0) \leqslant \text{val}_{G_{\text{sch}(S)}}(\sigma) = \sup_{v_0 \overset{w'}{\rightsquigarrow}_\sigma \text{ in } G_{\text{sch}(S)}} \text{val}(w') \leqslant \sup_{v_0 \overset{w}{\rightsquigarrow} \text{ in } G} \text{val}(w) = \text{val}_G(v_0). \qquad \square$$

The strategy which is used above to achieve a small value is not positional. However the existence of a positional strategy, which corresponds to a choice of $s \in S$, is guaranteed by assumption on val.

Lemma 3.5 (Adding edges consistent with a single choice)

Let $\sigma : \{S\} \to E$ be a uniform positional strategy for Eve in $G_{\text{sch}(S)}$, let $\sigma(S) = (S, c, s)$, and let G' be the graph over V obtained from G by adding all c-edges from vertices in $S \backslash \{s\}$ to s. Then for all vertices $v \in V$ it holds that

$$\text{val}_{G'}(v) \leqslant \text{val}_{G_{\text{sch}(S)}}(\sigma_v).$$

Again, the converse inequality is clear and irrelevant. This will of course be used when σ is optimal in which case the right-hand side is $\mathrm{val}_{G_{\mathrm{sch}(S)}}(v) \leqslant \mathrm{val}_G(v)$ by Lemma 3.4.

Proof. Let $v_0 \in V$ and let π' be a path in G'; it is of the form

$$\pi' : v_0 \overset{w_0}{\rightsquigarrow} v'_1 \xrightarrow{c} s \overset{w_1}{\rightsquigarrow} v'_2 \xrightarrow{c} s \overset{w_2}{\rightsquigarrow} \dots$$

for some (possibly empty, finite, or infinite) sequence of vertices $v'_1, v'_2, \dots$ satisfying

$$v_0 \overset{w_0}{\rightsquigarrow} v'_1 \text{ in } G \qquad \text{and} \qquad \forall j \geqslant 1, v'_j \in S \backslash \{s\} \text{ and } s \overset{w_j}{\rightsquigarrow} v'_{j+1} \text{ in } G$$

and where $|w_j| \geqslant 1$ for $j \geqslant 1$ since edges from s in G' belong to G. Again, if there are finitely many such vertices, it also holds that $v'_\ell \overset{w_\ell}{\rightsquigarrow}$ in G for the last index ℓ (or $v_0 \overset{w_\ell}{\rightsquigarrow}$ if the sequence is empty).

We let i_0 be equal to 1 if $|w_0| \geqslant 1$ and to 2 otherwise. Now for all $i \geqslant i_0$, we let v_i be the predecessor of v'_i in π', and we have

$$v_i \xrightarrow{c'_i} v'_i \in S \text{ in } G \qquad \text{thus} \qquad v_i \xrightarrow{c'_j} S \xrightarrow{c} s \text{ in } G_{\mathrm{sch}(S)},$$

and moreover the right-hand side is consistent with σ. Therefore the path

$$\pi : \begin{cases} v_0 \overset{w'_0}{\rightsquigarrow} v_1 \xrightarrow{c'_1} \ S \xrightarrow{c} s \overset{w'_1}{\rightsquigarrow} v_2 \xrightarrow{c'_2} S \xrightarrow{c} s \overset{w'_2}{\rightsquigarrow} \dots & \text{if } |w_0| \geqslant 1 \\ \qquad\qquad\quad S \xrightarrow{c} s \overset{w'_1}{\rightsquigarrow} v_2 \xrightarrow{c'_2} S \xrightarrow{c} s \overset{w'_2}{\rightsquigarrow} \dots & \text{if } |w_0| = 0, \end{cases}$$

where w'_j is obtained by removing the last letter of w_j, is consistent with σ in $G_{\mathrm{sch}(S)}$ and has the same coloration as π'. □

We may now prove Theorem 3.3.

Proof of Theorem 3.3. Let $S \subseteq V$ and consider the finite single choice arena $G_{\mathrm{sch}(S)}$ given by G, S and c. Note that it has size $|V| + 1$ and degree $\max(|V|, \deg(G))$ therefore Eve has a uniformly optimal positional strategy σ, and we let $s \in S$ be given by $\sigma(S) = (S, c, s)$.

It follows from Lemmas 3.4 and 3.5 that the graph G' obtained from G by adding all c-edges from $S \backslash \{s\}$ to s satisfies for all v,

$$\mathrm{val}_{G'}(v) \leqslant \mathrm{val}_G(v),$$

and therefore since G is c-saturated it must contain all these edges. □

3.2 Multiple choice arenas

As their name suggests, multiple choice arenas extend single choice arenas by forcing Eve to make many *consistent* choices simultaneously. The construction is nevertheless very similar, and Lemmas 3.6 and 3.7 proved below are direct analogues to Lemmas 3.4 and 3.5 from the previous subsection. Apart from introducing all subsets as Eve-vertices, we need to slightly simplify the main mechanic (this is required for Lemma 3.7), which now requires a c which is (not only weakly) good for Eve and ultimately good for Eve.

The *multiple choice arena* given by G and c is the arena G_{mch} over $V \cup \mathcal{P}(V)$ with $V_{\mathrm{Adam}} = V$ and $V_{\mathrm{Eve}} = \mathcal{P}(V)$ given by

$$\begin{aligned} v \xrightarrow{c'} v' \text{ in } G_{\mathrm{mch}} &\iff v \xrightarrow{c'} v' \text{ in } G \\ v \xrightarrow{c'} S \text{ in } G_{\mathrm{mch}} &\iff c' = c \text{ and } v \in S \\ S \xrightarrow{c'} v \text{ in } G_{\mathrm{mch}} &\iff c' = c \text{ and } v \in S. \end{aligned}$$

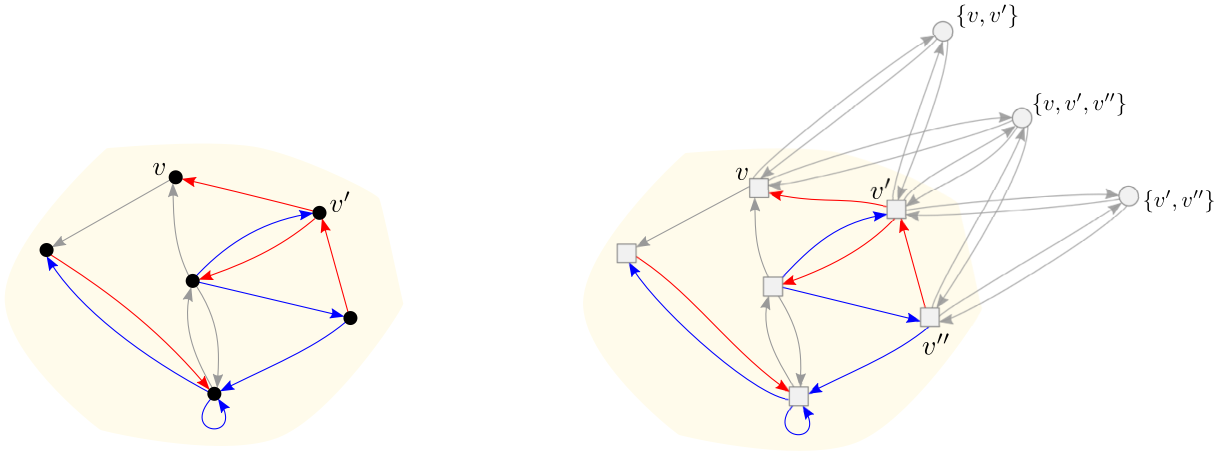


Figure 3.4: On the left, a $\{\text{blue}, \text{red}, \text{gray}\}$-arena G, and on the right, the multiple choice arena given by $c = \text{gray}$. For readability, only three of the 2^6 Eve-vertices are depicted.

An example is depicted in Figure 3.4.

We call vertices $S \in V_{\text{Eve}} = \mathcal{P}(V)$ *choice vertices*. Intuitively, Adam follows a path in G with the additional possibility at any point from a vertex $v \in V = V_{\text{Adam}}$ to enter a choice vertex $S \ni v$, seeing a c. It is then left to Eve to choose a successor from S, seeing another occurrence of c. A natural choice is to go back to v, which guarantees a small value by our assumption on c.

Since Adam can now ensure to see arbitrarily many consecutive c's (that we have introduced), or even infinitely many, we must now assume that the colour c we use is good for Eve and ultimately good for Eve.

Lemma 3.6 (Multiple choice arenas are easy)

For all $v \in V$ it holds that $\text{val}_{G_{\text{mch}}}(v) \leqslant \text{val}_G(v)$.

The proof is similar to that of Lemma 3.4.

Proof. Let $v_0 \in V$. Consider the (non-positional) Eve-strategy σ from v_0 satisfying

$$\sigma(\pi\ (v, c, S)) = (S, c, v),$$

for all finite paths π and all choice vertices $S \ni v$. In words, when arriving in S from v (necessarily with colour c) Eve picks the opposite edge.

Let π' be a path in G_{mch} consistent with σ. It is of the form

$$\pi' : v_0 \overset{w_0}{\rightsquigarrow} v_1 \xrightarrow{c} S_1 \xrightarrow{c} v_1 \overset{w_1}{\rightsquigarrow} v_2 \xrightarrow{c} S_2 \xrightarrow{c} v_2 \overset{w_2}{\rightsquigarrow} \dots$$

for some (possibly empty, finite, or infinite) sequence of vertices and subsets $v_1, S_1, v_2, S_2 \dots$ satisfying for all $i \geqslant 0$ that

$$v_i \overset{w_i}{\rightsquigarrow} v_{i+1} \text{ in } G \qquad \text{and} \qquad v_{i+1} \in S_{i+1}.$$

Moreover if the sequence is finite then its last index ℓ is such that $v_\ell \overset{w_\ell}{\rightsquigarrow}$ in G.

If $\sum_i |w_i|$ is bounded then $\mathrm{col}(\pi')$ is of the form uc^ω for some finite word u, thus $\mathrm{val}(\pi') = \min_{w \in C^\omega}(uw) \leqslant \mathrm{val}_G(v_0)$. Otherwise consider the path obtained by-passing the occurrences of choice vertices, formally

$$\pi : v_0 \overset{w_0}{\rightsquigarrow} v_1 \overset{w_1}{\rightsquigarrow} v_2 \overset{w_2}{\rightsquigarrow} \dots.$$

It defines an infinite path in G which moreover satisfies that $w' = \mathrm{col}(\pi')$ which is obtained from $w = \mathrm{col}(\pi)$ by adding c's. Hence we have $w' \leqslant w$ since c is good for Eve and we conclude as previously that

$$\mathrm{val}_{G_{\text{mch}}}(v_0) \leqslant \mathrm{val}_{G_{\text{mch}}}(\sigma) = \sup_{v_0 \overset{w'}{\rightsquigarrow}_\sigma \text{ in } G_{\text{mch}}} \mathrm{val}(w') \leqslant \sup_{v_0 \overset{w}{\rightsquigarrow} \text{ in } G} \mathrm{val}(w) = \mathrm{val}_G(v_0).$$

□

Now a uniform positional strategy for Eve in the multiple choice arena G_{mch} is given by

$$\begin{array}{rccl} \sigma : & \mathcal{P}(V) & \to & E_{G_{\text{mch}}} \\ & S & \mapsto & \sigma(S) = (S, c, s) \text{ with } s \in S, \end{array}$$

and for simplicity we abusively write $\sigma(S) = s \in S$, identifying uniform positional strategies in G_{mch} with choice functions $\mathcal{P}(V) \to V$. In contrast to the case of single choice arenas we may now simultaneously add a very large number of edges to G.

Lemma 3.7 (Adding edges consistent with a choice function)

Let $\sigma : \mathcal{P}(V) \to V$ be a uniform positional strategy for Eve in G_{mch} and let G' be obtained by adding to G all edges of the form $s' \xrightarrow{c} \sigma(S)$ for $s' \in S \subseteq \mathcal{P}(V)$. Then for all vertices $v \in V$ it holds that

$$\mathrm{val}_{G'}(v) \leqslant \mathrm{val}_{G_{\text{mch}}}(\sigma_v).$$

It is important for the proof below that the simpler "back-and-forth" mechanic is used. It is not clear whether the same result can be obtained by combining the mechanic of single choice arenas (which makes a weaker assumption on c) with multiple choices: the proof of Lemma 3.5 uses the fact that no edge outgoing from $s = \sigma(S)$ is added in G', which fails in the context of multiple choices.

Proof. Let $v_0 \in V$ and let π' be a path in G'. Then

$$\pi' : v_0 \overset{w_0}{\rightsquigarrow} v_1 \xrightarrow{c} \sigma(S_1) \overset{w_1}{\rightsquigarrow} v_2 \xrightarrow{c} \sigma(S_2) \overset{w_2}{\rightsquigarrow} \dots$$

for some (possibly empty, finite, or infinite) sequence of vertices and subsets $v_1, S_1, v_2, S_2 \dots$ satisfying

$$v_0 \overset{w_0}{\rightsquigarrow} \text{ in } G \qquad \text{and } \forall i \geqslant 1, v_i \in S_i \text{ and } \sigma(S_i) \overset{w_i}{\rightsquigarrow} v_{i+1}.$$

Once again, if there are finitely many such vertices, it also holds that $\sigma(S_\ell) \overset{w_\ell}{\rightsquigarrow}$ in G for the last index ℓ (or $v_0 \overset{w_0}{\rightsquigarrow}$ if the sequence is empty).

Then

$$\pi : v_0 \overset{w_0}{\rightsquigarrow} v_1 \xrightarrow{c} S_1 \xrightarrow{c} \sigma(S_1) \overset{w_1}{\rightsquigarrow} v_2 \xrightarrow{c} S_2 \xrightarrow{c} \sigma(S_2) \overset{w_2}{\rightsquigarrow} \dots$$

defines a path in G_{mch} which is consistent with σ, and whose colour is obtained by adding c's to $\mathrm{col}(\pi)$. Hence

$$\mathrm{val}(\pi') \leqslant \mathrm{val}(\pi) \leqslant \mathrm{val}_{G_{\text{mch}}}(\sigma_{v_0}),$$

and as usually the result follows by taking a supremum. □

Lemmas 3.6 and 3.7 together precisely state that one may add sufficiently many c-edges to G without increasing val, provided there is an optimal Eve strategy over the multiple choice arena, which corresponds to the cardinality hypotheses in Theorem 3.2. Since moreover the given letter is assumed to be good for Adam, we conclude with the theorem by applying Corollary 3.3.

Conclusion and perspectives for Part I

We have introduced well-monotonic graphs, and shown that they characterise positionality over arbitrary arenas, in the presence of a strongly neutral letter. We have also discussed many examples, illustrating the modularity of our approach. In particular, we could establish a novel closure property for positional objectives by studying a natural construction over well-monotonic graphs. Such a proof technique is novel, and we believe that it provides (much needed) handles for attacking Kopczyński's conjecture (closure under union for prefix-closed positional objectives). We now (informally) discuss two concrete directions on this front, and also discuss other perspectives.

Extending K-monotonicity. Kopczyński proposed to study monotonic conditions (which we will call K-monotonic to avoid ambiguity), which he showed to be positionally determined and closed under unions. We explain how this notion instantiates to our setting.

Given an ordinal α, we let $\mathcal{L}_\alpha^\varnothing$ denote the unique $\varnothing$-graph over α (it has no edge). Then we claim (without proof) that K-monotonic conditions can be understood as those which admit (non-uniform) well-monotonic universal graphs of the form $\mathcal{L}_\alpha^\varnothing \otimes \mathcal{L}_0$, where $\mathcal{L}_0$ is a fixed well-monotonic graph (see Figure 3.5 for an illustration). An example is the co-Büchi condition: the construction presented in Chapter 2 is of this form, with $\mathcal{L}_0$ is the $\{\text{safe}, \text{bad}\}$-graph comprised of a single vertex with a safe-loop.

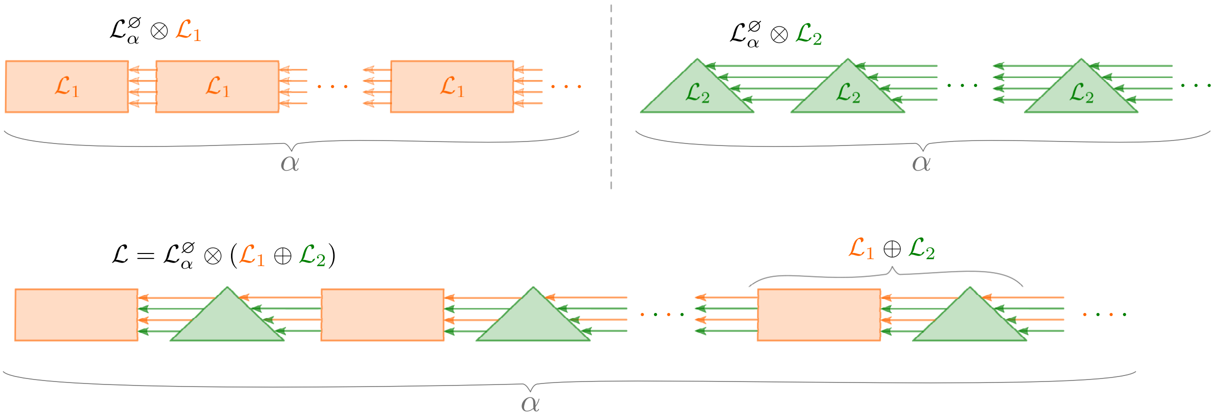

$\mathcal{L}_\alpha^\varnothing \otimes \mathcal{L}_1$
$\mathcal{L}_1$
$\mathcal{L}_1$
$\mathcal{L}_1$
α
$\mathcal{L}_\alpha^\varnothing \otimes \mathcal{L}_2$
$\mathcal{L}_2$
$\mathcal{L}_2$
$\mathcal{L}_2$
α
$\mathcal{L} = \mathcal{L}_\alpha^\varnothing \otimes (\mathcal{L}_1 \oplus \mathcal{L}_2)$
$\mathcal{L}_1 \oplus \mathcal{L}_2$
α

Figure 3.5: An illustration of the monotonic graphs corresponding to K-monotonic conditions and their interleaving.

Closure under finite unions is then naturally supported, as suggested by Kopczyński's proof, by simply interleaving (see Figure 3.5) the building blocks, formally setting $\mathcal{L} = \mathcal{L}_\alpha^\varnothing \otimes (\mathcal{L}_1 \oplus \mathcal{L}_2)$ where $\oplus$ is obvious (a formal definition is also given in Chapter 8). Can we better understand this fact? Can we generalise a similar easy construction to larger classes of valuations (which, ideally, have would have nice characterisations)?

Colour-disjoint unions. Taking a small step back, what we did for lexicographical products can be seen as

(i) formalising parity games as (colour-disjoint) products of Co-Büchi conditions, and then

(ii) extending Emerson and Jutla's inductive universality proof [EJ91] from co-Büchi to arbitrary (prefix-independent) well-monotonic graphs as building blocks.

Now observe that the Rabin condition is precisely a colour-disjoint union of co-Büchi conditions, when these are defined in general by

$$\bigsqcup_{p=1}^{h} W_p = \{w \in C^\omega \mid \exists p, w_p \text{ is infinite and } w_p \in W_p\},$$

where $C = \bigsqcup_{p=1}^{h} C_p$. Therefore we have (i). Fortunately, Klarlund and Kozen [KK91] give an analogous (much more involved) template for Rabin progress measures, providing the needed basis (and well grounded hope) for (ii). Can we complete the proof of (ii), confirming Kopczyński's conjecture?

Positionality over finite arenas. The case of bi-positionality indicates that there are many more valuations which are positional when restricting to finite arenas. Notorious examples include mean-payoff and discounted games, whose positionality eludes the well-founded strategy-folding technique (Theorem 1.1).

Still, universal monotonic graphs are available for threshold mean-payoff games over finite arenas (Corollary 2.3) via energy games, and likewise for discounted games. Actually, the finite structuration result (Theorem 3.1) gives such a construction whenever there is a weakly neutral colour (which is the case of threshold mean-payoff, but not (threshold) discounted valuations).

We have shown that well-foundedness in (universal) monotonic graphs is a structural implementation of the strategy-folding technique, and in some way this technique appears to be *complete* for positionality over arbitrary arena, however not adapted to finite arenas. Can we find such a structural implementation (in monotonic graphs) of techniques which are adapted to finite arenas? In this hopeful scenario, can we characterise valuations which admit such universal monotonic graphs?

An easy (but not uninteresting) instantiation of this framework is suggested by Shapley's technique (based on Banach's theorem) and the case of discounted games: if fixpoints can be guaranteed to be unique, then positionality (for both players) follows. Remarkably, Kozachinskiy [Koz21a] essentially proved that this technique is complete for *continuous* valuations $A^\omega \to \mathbb{R}$ which are bi-positional over finite arenas.

Another very promising candidate technique is given by Ehrenfeucht and Mycielski's cyclic games [EM79]. Aminof and Rubin [AR17] have given sufficient conditions for the technique to apply to a given objective W. Can we find a structural counterpart (over corresponding monotonic graphs), and apply the proposed framework?

Part II

Finite monotonic graphs and value iterations

Introduction

For the remainder of the thesis, the focus is on designing algorithms for solving games by relying on finite monotonic graphs. The main insight we build on, which is not hard to prove (see Theorem 4.3), is that when given a $(\mathcal{C}, \mathrm{val})$-universal monotonic $\mathcal{L}$, one may reduce the problem of solving a val-game over an arena $G \in \mathcal{C}^{\mathrm{ar}}$ to the computation of the $\mathcal{L}$-evaluation ψ of G – which we recall to be defined as a least fixpoint over the space L^V of progress measures.

Generally, we will choose the class of graphs $\mathcal{C}$ to consist of all graphs of cardinality $\leqslant n$. For simplicity, we will say that $\mathcal{L}$ is (n, val)-universal, or n-universal when val is clear from context, if it is val-universal for all graphs of cardinality $\leqslant n$. We also recall (Lemma 1.8) that in the case where val corresponds to a prefix-independent objective W, a monotonic graph $\mathcal{L}$ is n-universal if it satisfies W and embeds all graphs of size $\leqslant n$ which satisfy W.

Value iteration algorithms. Assuming operations relative to $\mathcal{L}$ (essentially, access to the min-predecessor table ρ) can be carried out in constant time, $\mathrm{Upd}(\phi)$ can be computed in $O(m)$ from ϕ. Therefore, ψ can be computed by Kleene iteration with runtime $O(mn|L|)$ and space $O(n \log |L|)$. Using a standard technique inherited from the resolution of safety games, on may reduce the runtime to $O(m|L|)$ in general.

This simple algorithm belongs to the general paradigm of *value iterations*. Chatterjee and Henzinger [CH08] presented a survey of value iteration algorithms with a broad (informal) definition, applying to computations of nested fixpoints and to the more general settings of concurrent and stochastic games. We prefer to work with the more specific (and formal) definition via finite monotonic graphs, sacrificing over some generality which is not needed for us.

Part II is only concerned with value iteration algorithms. Designing an efficient value iteration algorithm amounts to exhibiting small universal graphs, and therefore the focus here will be only on the size of $\mathcal{L}$. We will always assume that the valuation under study is positional and admits a weakly neutral letter, and therefore our finite structuration result (Theorem 3.1) implies that monotonicity is not restrictive: universal graphs of minimal size can be chosen monotonic.

Our work in this second part has a significant bibliographical component; our purpose is often to give an alternative presentation of recent elements of the literature in the unifying vocabulary of monotonic graphs. In many important cases, the state-of-the-art runtime bound is given by such an approach. However, these are not practical: the worst-case upper bound $O(mn|L|)$ is generally met with simple (size one) examples, and an exponential behaviour over practical (or random) instances is generally observed.

Organisation of Part II. We start in Chapter 4 by relating our approach with the separating technique of Bojańczyk and Czerwiński [BC18] by showing that *strongly* separating automata can be determinised or turned into universal monotonic graph with no blow-up. This essentially follows from the finite structuration result (Theorem 3.1) and requires a weakly neutral letter. A similar result was established in [CDF+18] in the case of parity games. We also introduce, discuss, and compare two generic variants of value iteration, concluding that they are essentially equivalent.

Chapter 5 studies the case of parity games. We show that in this case, saturated graphs correspond to ordered trees, and hence universality of such graphs reduces to tree universality. We present the quasipolynomial construction of Jurdiński and Lazić [JL17], and the (almost) matching lower bound of Fijalkow [Fij18].

We then study threshold mean-payoff games in 6. As we have seen (Corollary 2.3), the connection with energy games immediately leads to an n-universal graph of size $(n-1)N$, and thus a value iteration algorithm with state-of-the-art quasipolynomial runtime $O(mnN)$. This corresponds exactly to the BCDGR algorithm [BCD+11]. We give a lower bound of $O(N^{1-1/n})$ on the size of n-universal graphs and a matching construction, up to a polynomial factor.

From the observation that there are restricted sets of weights (namely, $A = \{(-n)^p \mid p \in [0, d]\}$, which corresponds to parity games) for which there are quasipolynomial universal graphs, one could hope that such a result hold whenever the cardinality k of A is bounded. We show that this is not the case, by giving a $\Omega(n^{k-1})$ lower bound. Our informal conclusion for this chapter is that value iteration algorithms cannot significantly improve the state of the art for mean-payoff games.

Chapter 7 is concerned with mean-payoff parity games. We show how the construction of Daviaud, Jurdziński and Lazić [DJL18] instantiates to the vocabulary of monotonic graphs, and we reobtain their main result. We believe that the universality proof is of independent interest.

Finally, Chapter 8 studies multi mean-payoff games, in the $\limsup$ semantics for which they are concave in the sense of Kopczyński. We show that in this case, one may succinctly combine universal graphs for mean-payoff games. This leads to a value iteration algorithm with runtime $O(mdn \log(n) N)$, saving a factor $n/\log(n)$ from the approach of [VCD+15].

In a brief conclusion we collect a few questions which are left open, and future directions for value iterations.

All of Part II was derived in collaboration with Nathanaël Fijalkow. The results on value iteration, saturation (including Theorem 3.1 from Part I) and parity games are also joint with Thomas Colcombet. Results on mean-payoff games and universality (Chapter 6) are also the fruit of discussions with Paweł Gawrychowski, and were published in [FGO20].

Chapters 7 and 8 are (loosely) based on a joint work with Ashwani Anand, Nathanaël Fijalkow, Aliénor Goubault-Larrecq and Jérôme Leroux [AFGL+21] in which we proved similar results by using separating automata.

A journal version compiling roughly all results of Part II and co-authored with Thomas Colcombet, Nathanaël Fijalkow and Paweł Gawrychowski is under currently under review, and available at [CFG+21].

Separating automata and value iterations

4

Our purpose in this chapter is twofold: connecting universal monotonic graphs to Bojańczyk and Czerwiński's separating automata [BC18] (which were introduced for parity game but immediately instantiate to any positional objective), and introducing a generic framework for value iteration algorithms.

In the first section, we take a small detour and study valuations induced by monotonic graphs $\mathcal{L}$, establishing that these correspond to evaluations whenever $\mathcal{L}$ is finite. This allows us in Section 2 to show that strongly separating automata can be turned into universal monotonic graphs with no blow up, and also determinised. Section 3 introduces two different generic variants of value iterations in monotonic graphs, and compares their runtime.

1 From finite monotonic graphs to valuations

Fix a completely well-monotonic graph $\mathcal{L}$ over L. We have seen that monotonic graphs are colouration-monotonous thanks to left composition: if $\ell \geqslant \ell'$ then ℓ has more colourations than ℓ'. Moreover, we have seen that the maximal element $\top \in L$ has all c-loops, and therefore all colourations. This naturally suggests the definition of a valuation associated to $\mathcal{L}$, given by

$$\begin{array}{rccl} \mathrm{val}^{\mathcal{L}} : & C^\omega & \to & L \\ & w & \mapsto & \min\{\ell \in L \mid \ell \overset{w}{\rightsquigarrow} \text{ in } \mathcal{L}\}. \end{array}$$

Note that we have by right composition

$$\mathrm{val}^{\mathcal{L}}(w) \leqslant \ell \qquad \iff \qquad \forall n \in \omega, \quad \ell \overset{w_{\leqslant n}}{\rightsquigarrow} \bot,$$

therefore $(\mathrm{val}^{\mathcal{L}})^{\leqslant \ell}$ is a countable intersection of open subsets of C^ω hence $\mathrm{val}_{\mathcal{L}}$ is determined in general by Martin's theorem (and it is topologically weaker than for instance the parity objective).

Given a finite arena G, we have thus defined two different maps from $V \to L$: on one hand the $\mathrm{val}^{\mathcal{L}}$-values over G and on the other the $\mathcal{L}$-evaluation of G. These have different nature, the former is defined via a game whereas the latter is defined as a fixpoint. In particular, they do not coincide in general even over finite graphs; two examples are discussed in Figure 4.1.

However these two notions do coincide over all arenas as soon as $\mathcal{L}$ is finite.

Valuation versus evaluation. We fix a completely well-monotonic graph $\mathcal{L}$ and an arbitrary arena G. Recall from Chapter 1 that a uniform positional strategy σ can be obtained from the evaluation ψ of G in $\mathcal{L}$, simply by following from $v \in V_{\text{Eve}}$ an edge $v \xrightarrow{c} v'$ which minimises $\rho(\psi(v'), c)$. We

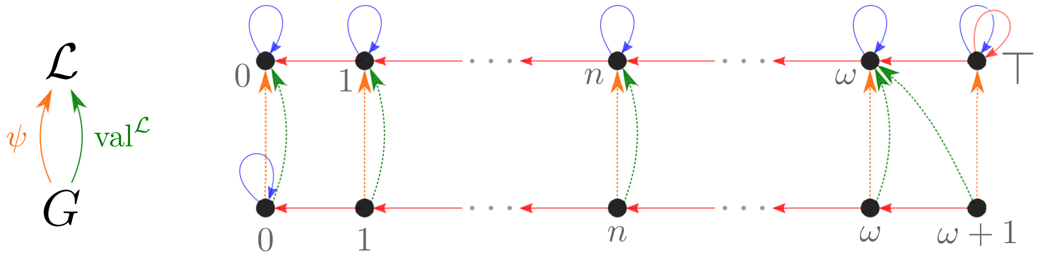


Figure 4.1: An example where $\mathrm{val}^{\mathcal{L}}$ and ψ do not coincide.

proved in Lemma 1.6 (second ingredient for Theorem 1.1) that finite paths $v \overset{w}{\rightsquigarrow}_{\sigma} v'$ in G satisfy that $\psi(v) \overset{w}{\rightsquigarrow} \psi(v')$ is a path in $\mathcal{L}$ and therefore it holds in general that for all v,

$$\mathrm{val}^{\mathcal{L}}_G(v) \leqslant \mathrm{val}^{\mathcal{L}}_G(\sigma) \leqslant \psi^{\mathcal{L}}_G(v).$$

The converse inequality holds if $\mathcal{L}$ is finite.

Theorem 4.1 (Identity over finite $\mathcal{L}$)

If $\mathcal{L}$ is finite then $\mathrm{val}^{\mathcal{L}}_G$ and $\psi^{\mathcal{L}}_G$ coincide. In particular, $\mathrm{val}^{\mathcal{L}}$ is uniformly positionally determined over all arenas.

Proof. We will show that $\mathrm{val}^{\mathcal{L}}_G$ is a prefixpoint of Upd_G, which concludes by Knaster-Tarski since ψ is its smallest prefixpoint. Note that since $\mathcal{L}$ is finite, any infimum or supremum is met over L.

We first show that for any $w' \in C^{\omega}$ and $c \in C$ we have

$$\rho(\mathrm{val}^{\mathcal{L}}(w'), c) \leqslant \mathrm{val}^{\mathcal{L}}(cw').$$

Indeed, we have

$$\begin{aligned} \mathrm{val}^{\mathcal{L}}(w') &= \max\nolimits_{n \in \mathbb{N}} \min\{\ell \in L \mid \ell \overset{w'_{\leqslant n}}{\rightsquigarrow} \bot \text{ in } \mathcal{L}\} \\ &= \min\{\ell \in L \mid \ell \overset{w'_{\leqslant n_0}}{\rightsquigarrow} \bot \text{ in } \mathcal{L}\} = \ell'_0, \end{aligned}$$

for some $n_0 \in \mathbb{N}$ and $\ell'_0 \in \mathcal{L}$. Now let $\ell \in \mathcal{L}$ be such that $\ell \overset{cw'_{\leqslant n_0}}{\rightsquigarrow} \bot$ in $\mathcal{L}$. Then for some ℓ', $\ell \xrightarrow{c} \ell' \overset{w'_{\leqslant n_0}}{\rightsquigarrow} \bot$ therefore $\ell' \geqslant \ell'_0$ and thus by monotonicity $\ell \geqslant \rho(\ell', c) \geqslant \rho(\ell'_0, c)$. Now we have $\mathrm{val}^{\mathcal{L}}(cw') \geqslant \ell \geqslant \ell_0$ which concludes with the wanted inequality.

- Let $v \in V_{\mathrm{Eve}}$, let σ be an optimal Eve strategy from v, let $\sigma(\varepsilon) = (v, c, v') = e$ and let σ' be the prestrategy from v' given by $\sigma'(\pi') = \sigma(e\pi)$. It is straightforward to check that $v \overset{w}{\rightsquigarrow}_{\sigma}$ in G if and only if $w = cw'$ and $v' \overset{w'}{\rightsquigarrow}_{\sigma'}$ in G. Therefore σ' is indeed a strategy and we obtain

$$\begin{aligned} \mathrm{val}^{\mathcal{L}}_G(v) = \mathrm{val}^{\mathcal{L}}(\sigma) = \max_{v \overset{w}{\rightsquigarrow}_{\sigma}} \mathrm{val}^{\mathcal{L}}(w) = \max_{v' \overset{w'}{\rightsquigarrow}_{\sigma'}} \mathrm{val}^{\mathcal{L}}(cw') \geqslant \\ \geqslant \max_{v' \overset{w'}{\rightsquigarrow}_{\sigma'}} \rho(\mathrm{val}^{\mathcal{L}}(w'), c) = \rho(\mathrm{val}^{\mathcal{L}}(\sigma'), c) \geqslant \rho(\mathrm{val}^{\mathcal{L}}(v'), c) \geqslant \mathrm{Upd}(\mathrm{val}^{\mathcal{L}}_G)(v). \end{aligned}$$

- Let $v \in V_{\text{Adam}}$, let $v \xrightarrow{c} v'$ in G, let τ be an optimal Adam strategy from v', and let τ be the prestrategy from v defined by $\tau(\pi) = \tau((v, c, v')\pi')$. Again, $v \overset{w}{\rightsquigarrow}_\tau$ in G if and only if $w = cw'$ and $v' \overset{w'}{\rightsquigarrow}_{\tau'}$ and thus τ defines a strategy from v. We now obtain similarly
$$\begin{aligned} \mathrm{val}^{\mathcal{L}}_G(v) \geqslant\ & \mathrm{val}^{\mathcal{L}}(\tau) = \max_{v \overset{w}{\rightsquigarrow}_\tau} \mathrm{val}^{\mathcal{L}}(w) = \max_{v' \overset{w'}{\rightsquigarrow}_{\tau'}} \mathrm{val}^{\mathcal{L}}(cw') \geqslant \\ & \geqslant \max_{v' \overset{w'}{\rightsquigarrow}_{\tau'}} \rho(\mathrm{val}^{\mathcal{L}}(w'), c) = \rho(\mathrm{val}^{\mathcal{L}}(\tau'), c) = \rho(\mathrm{val}^{\mathcal{L}}_G(v'), c) \end{aligned}$$
and the result follows by maximizing over outgoing edges $v \xrightarrow{c} v'$ in G. □

For convenience especially in the context of prefix-independent objectives, we also define $\mathrm{val}^{\mathcal{L}}$ when $\mathcal{L}$ is not complete by considering the completion $\mathcal{L}^\top$ of $\mathcal{L}$ which is given by
$$\mathrm{val}^{\mathcal{L}}(w) = \mathrm{val}^{\mathcal{L}^\top}(w) = \begin{cases} \top & \text{if there is no } w\text{-path in } \mathcal{L} \\ \min\{\ell \in \mathcal{L} \mid \ell \overset{w}{\rightsquigarrow} \text{ in } \mathcal{L}\} & \text{otherwise.} \end{cases}$$

Since all finite linear orders are well-orders, a finite well-monotonic graph is simply a finite monotonic graph. Theorem 4.1 states that solving the $\mathrm{val}^{\mathcal{L}}$-game if $\mathcal{L}$ is finite is equivalent to computing the $\mathcal{L}$-evaluation.

2 Determinisation of strongly separating automata

n-approximations and n-universality. Fix an arbitrary winning condition $W \subseteq C^\omega$. For $n \in \omega$, we define the *n-approximation* $W_n \subseteq C^\omega$ of W to be the set of colourations of infinite paths from vertices satisfying W in graphs of size $\leqslant n$.

Lemma 4.1 (Rephrasing n-universality for finite monotonic graphs)

A finite monotonic graph $\mathcal{L}$ over L is (n, W)-universal if and only if there is a vertex $\ell_0 \in L$ such that ℓ_0 satisfies W and has all colourations from W_n in $\mathcal{L}$.

Proof. Assume $\mathcal{L}$ is (n, W)-universal, let ℓ_0 be the largest vertex in L which satisfies W and let $w \in W_n$: there is a graph G over V of size $|V| \leqslant n$ and a vertex $v \in V$ satisfying W with a path $v \overset{w}{\rightsquigarrow}$. Consider the evaluation ψ of G in $\mathcal{L}$.

Since $\mathcal{L}$ is (n, W)-universal ψ is W-preserving, therefore $\psi(v)$ satisfies W in $\mathcal{L}$ thus $\psi(v) \leqslant \ell_0$. Now w is a colouration from $\psi(v)$ in $\mathcal{L}$ and by colouration-monotonicity, it is also a colouration from ℓ_0.

Conversely, assume that there exists $\ell_0 \in L$ which satisfies W and has all colourations from W_n in $\mathcal{L}$ and let G be a graph of size $\leqslant n$. Consider the evaluation ψ of G in $\mathcal{L}$, which is a morphism in general (see Chapter 1); we show that it is W-preserving.

Let $v \in G$ and assume that v satisfies W. By Theorem 4.1 $\psi(v)$ coincides with $\mathrm{val}^L_G(v)$ which rewrites as
$$\psi(v) = \min\{\ell \in \mathcal{L} \mid v \overset{w}{\rightsquigarrow} \text{ in } G \implies \ell \overset{w}{\rightsquigarrow} \text{ in } \mathcal{L}\}.$$
Since G has size $\leqslant n$ and v satisfies W, all colourations from v in G belong to W_n and therefore $\psi(v) \leqslant \ell_0$ by minimality. Hence $\psi(v)$ satisfies W by colouration-monotonicity in $\mathcal{L}$ since ℓ_0 does, which concludes. □

Determinisation of finite monotonic graphs. A graph is *deterministic* if for every vertex v and every colour c, v has at most one c-successor. There is a natural way of making a *finite* monotonic graph $\mathcal{L}$ deterministic without increasing its size: from vertex v with colour c keep only the maximal successor of v (if there is any). This is not possible in general (see for instance the construction for Büchi games in Chapter 2) if $\mathcal{L}$ is not finite, since maximal successors may not be well defined.

Formally, for a finite monotonic graph $\mathcal{L}$ over L, we let $\det(\mathcal{L})$ be the graph over L given by

$$\ell \xrightarrow{c} \ell' \text{ in } \det(\mathcal{L}) \qquad \iff \qquad \ell' = \max \Delta(\ell, c).$$

Note that $\Delta(\ell, c)$ may be empty, in which case ℓ has no c-successor in $\det(\mathcal{L})$ since the maximum is not defined. It is easy to see that $\det(\mathcal{L})$ is indeed a graph: if ℓ were a sink in $\det(\mathcal{L})$ then it would have no c-successor in $\mathcal{L}$ for any c and it would therefore be a sink in $\mathcal{L}$ which is excluded.

Note that $\det(\mathcal{L})$ is a subgraph of $\mathcal{L}$, stated differently the identity defines a graph-morphism from $\det(\mathcal{L})$ to $\mathcal{L}$. The following lemma states that one does not lose any colouration when going from $\mathcal{L}$ to $\det(\mathcal{L})$.

Lemma 4.2 (No fewer colours in $\det(\mathcal{L})$)

The identity morphism from $\det(\mathcal{L})$ to $\mathcal{L}$ is colouration-preserving.

Proof. Let $\pi : \ell_0 \xrightarrow{c_0} \ell_1 \xrightarrow{c_1} \dots$ be an infinite path in $\mathcal{L}$. We construct by induction a path $\pi' : \ell'_0 \xrightarrow{c_0} \ell'_1 \xrightarrow{c_1} \dots$ from $\ell'_0 = \ell_0$ in $\det(\mathcal{L})$ with the same colouration $c_0 c_1 \dots$ and such that for all i, $\ell'_i \geqslant \ell_i$. The initialisation is trivial, and assume constructed π' up to ℓ'_i.

Since $\ell'_i \geqslant \ell_i$ we have by left composition in $\mathcal{L}$ that $\Delta(\ell'_i, c_i) \supseteq \Delta(\ell_i, c_i)$. Therefore $\Delta(\ell'_i, c_i)$ is non-empty since ℓ_{i+1} belongs to $\Delta(\ell_i, c_i)$ and $\ell'_{i+1} = \max \Delta(\ell'_i, c_i)$ is larger than ℓ_{i+1} and such that $\ell'_i \xrightarrow{c_i} \ell'_{i+1}$, concluding the induction and the proof. □

Separating automata. An (n, W)-*separating automaton* is a finite graph with a vertex v_0 which has all colourations from W_n and excludes all colourations from $({}^{c}W)_n$. Such an automata is *strongly separating* if v_0 even excludes all colourations from ${}^{c}W$, or stated differently v_0 satisfies W. This is illustrated in Figure 4.2.

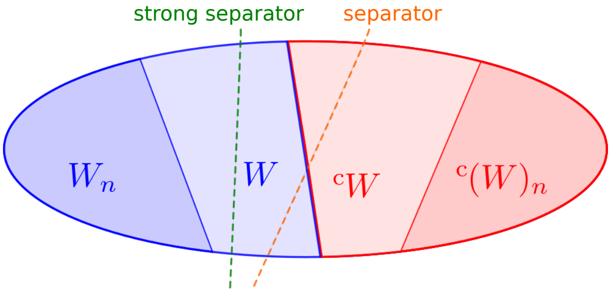


Figure 4.2: An illustration of the definition of separators. The universe is the set C^ω of infinite words.

Bojańczyk and Czerwiński [BC18] introduced separating automata specifically for parity games, and showed that a *deterministic* separating automaton $\mathcal{A}$ allows to reduce a game G to an equivalent safety game $G \rhd A$ of size $n_G n_{\mathcal{A}}$ and with $m_G n_{\mathcal{A}}$ edges. Intuitively, $G \rhd \mathcal{A}$ is played just like G where additionally, colours are read in $\mathcal{A}$ from v_0; the bad colour (which corresponds to Eve losing) is seen when no transition exist in $\mathcal{A}$ with the current colour. If W is positionally determined, then

G and $G \rhd \mathcal{A}$ are equivalent; we refer to [BC18] (see also [CFG+21]) for formal details which will be omitted here.

For parity games, it was established in [BC18; CDF+18; Par20] that the four earlier quasipolynomial algorithms from [CJK+17; FJS+17; JL17; Leh18] all correspond to constructions of (nondeterministic) strongly separating automata. To date, no construction of a (quasipolynomial) nondeterministic separating automaton which is not strongly separating is known. We note however that the quasipolynomial lower bound established from universal trees [CDF+18] or equivalently universal (monotonic) graphs [CF18; CFG+21], applies only to *strongly* separating automata: there might exist (weakly) parity-separating automata of polynomial size. Finding such an automaton, if it is moreover deterministic (or at least, good-for-small-games as in [Par20] or [CFG+21]), would lead to a polynomial algorithm for parity games. The determinisation result below, as well as the connection to (inherently asymmetric) monotonic graphs, holds only for strong separators.

Determinisation of strongly separating automata. We actually already have all the tools in hands to determinise separating automata.

Theorem 4.2 (Determinisation of strongly separating automata)

Assume that W has a weakly neutral colour and is uniformly positionally determined over finite arenas, let $n \in \omega$ and let $\mathcal{A}$ be a (n, W)-separating automaton. There exists a finite monotonic graph $\mathcal{L}$ no larger than $\mathcal{A}$ which is (n, W)-universal. In particular, $\det(\mathcal{L})$ is a deterministic (n, W)-separating automaton.

Streamlining the application of the finite saturation result (Theorem 3.1), $\mathcal{L}$ is obtained from $\mathcal{A}$ by first saturating with respect to the neutral colour, then closing around other edges, and finally quotienting with respect to $\xrightarrow{0}$.

Proof. We simply let $\mathcal{L}$ be a W-structuration of $\mathcal{A}$, obtained via Theorem 3.1. Then $v_0 \in A$ has all colourations from W_n in $\mathcal{A}$ therefore also in $\mathcal{L}$, and its image in $\mathcal{L}$ satisfies W by W-preservation, so $\mathcal{L}$ is (n, W)-separating thanks to Lemma 4.1. The last claim then follows from Lemma 4.2. □

Therefore, we will only concentrate on solving games when given a family of n-universal finite monotonic graphs, which have much more structure than arbitrary deterministic separating automata. We note however that the complexity of solving the safety game $G \rhd \mathcal{A}$ is $O(m_{G \rhd \mathcal{A}}) = O(mn_{\mathcal{A}})$, which matches the complexity of the value iteration approach. Actually, it is not hard to see that the value iteration algorithm is nothing but a symbolic implementation of standard (Kleene iteration) algorithms for solving the product safety game $G \rhd \mathcal{L}$, saving on space complexity by taking advantage of the linear order over $\mathcal{L}$. This was already observed by Bernet, Janin and Walukiewicz [BJW02] for parity games, and later by Brim, Chaloupka, Doyen, Gentilini and Raskin for energy games [BCD+11]; full details in our generic setting can be found in [CFG+21], but we omit them here.

3 Value iterations

We fix a C-valuation val which is assumed to be positionally determined over finite arenas, an arena G over V of size $|V| = n$ and a finite completely monotonic graph $\mathcal{L}$ over L which should be thought as being (much) larger than n (for instance quasipolynomial or exponential). We let $t_{\mathcal{L}}$

denote the required runtime for the two *elementary operations* in $\mathcal{L}$: comparing two vertices, and computing $\rho(\ell', c)$.This is usually polynomial in n. We have the following result.

Theorem 4.3 (Solving G by evaluation)

If $\mathcal{L}$ is n-universal with respect to val, then for all $v \in V$ we have $\mathrm{val}_G(v) = \mathrm{val}_{\mathcal{L}}(\psi_G(v))$. In particular, if val is a qualitative valuation given by $W \subseteq C^\omega$, then $v \in V$ is winning if and only if $\psi_G(v)$ satisfies W in $\mathcal{L}$.

Since this easy but important result holds even for infinite $\mathcal{L}$, we prefer to prove it directly rather than via Theorem 4.1.

Proof. Given a uniform positional strategy σ for Eve in G and a progress measure $\phi : V \to L$, we have in general

$$\mathrm{Upd}_G(\phi) \leqslant \mathrm{Upd}_{G_\sigma}(\phi).$$

Indeed, recall that by definition,

$$\mathrm{Upd}_G(\phi) = \begin{cases} \min\limits_{v \xrightarrow{c} v' \text{ in } G} \rho(\phi(v'), c) \text{ if } v \in V_{\mathrm{Eve}} \\ \max\limits_{v \xrightarrow{c} v' \text{ in } G} \rho(\phi(v'), c) \text{ if } v \in V_{\mathrm{Adam}}, \end{cases}$$

which is smaller in G than in G_σ: nothing changes for $v \in V_{\mathrm{Adam}}$ while the min is restricted for $v \in V_{\mathrm{Eve}}$. Therefore $\psi_G \leqslant \psi_{G_\sigma}$ in general. Now if σ is chosen to be positional and optimal from a given vertex v, then we have $\mathrm{val}_G(v) = \mathrm{val}_{G_\sigma}(v)$ and thus by val-preservation $\psi_{G_\sigma}(v)$ has value $\mathrm{val}_G(v)$ in $\mathcal{L}$ and therefore $\mathrm{val}_{\mathcal{L}}(\psi_G(v)) \leqslant \mathrm{val}_G(v)$ by colouration monotonicity.

Moreover, this inequality cannot be strict otherwise by Lemma 1.6 we would have a (positional) strategy achieving a better value. □

Hence to solve the W-game over G it suffices to compute the $\mathcal{L}$-evaluation ψ; the winning region is the set of vertices whose image satisfies W in $\mathcal{L}$, which rephrases as being $\leqslant \ell_0$ where ℓ_0 is the maximal vertex satisfying W in $\mathcal{L}$. In all further applications, W is prefix-increasing, therefore $\mathcal{L}$ can be taken of the form $\mathcal{L}^\top$ where $\mathcal{L}$ satisfies W (see Lemma 1.8), and the above condition rewrites as $\psi(v) < \top$.

3.1 Local and global value iterations

The simplest way of computing the evaluation ψ of G is by Kleene iteration: start with the minimal progress measure $\phi_0 = \phi_\perp : V \to L$ which assigns to each vertex v the minimal element $\perp = \min L$, and iteratively compute $\phi_{i+1} = \mathrm{Upd}(\phi_i)$ until reaching the least fixpoint. The number of iterations $\mathrm{it}_{\mathrm{glob}}$ is upper bounded by $n|L|$ since in each iteration at least one vertex has a strict increase. In a naive implementation, computing ϕ_{i+1} from ϕ_i requires runtime $O(mt_{\mathcal{L}})$, which gives a first worst-case runtime upper bound of $O(nmt_{\mathcal{L}}|L|)$.

In the case of a safety game (see Chapter 2) both $t_{\mathcal{L}}$ and $\mathcal{L}$ can be taken constant, therefore this gives runtime $O(nm)$. It is folklore that safety games can be solved in linear time $O(m)$ with two slightly different enhanced implementations of the Kleene iteration, which we will call the *local* and *global* iterations (see Figure 4.3).

Adapting the algorithms from safety games to the more general fixpoint iteration described above yields two simple variants of the *value iteration algorithm* both of which have worst-case runtime

$$O(mt_{\mathcal{L}}|L|),$$

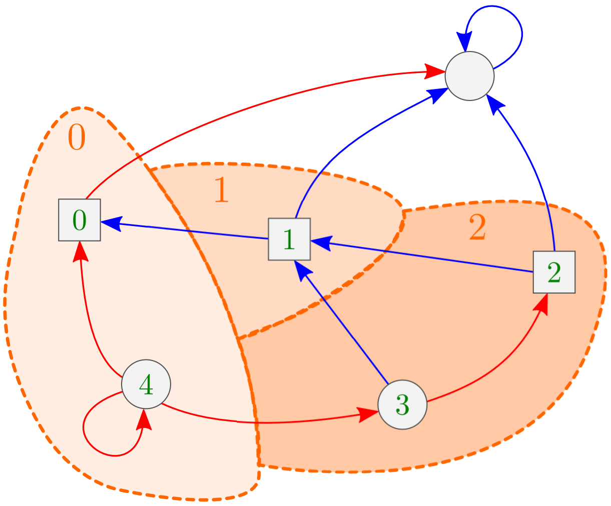


Figure 4.3: An example of a safety game with a single winning vertex (at the top). Blue edges are safe and red edges are bad. Two different algorithms for solving it are illustrated.
The global iteration, in orange, corresponds to the natural Kleene iteration: at the i-th iteration, vertices from which Adam can ensure to see a bad-edge in i step are added to his winning region.
The local iteration corresponds to a variant in which vertices are added non-deterministically to Adam's winning region until it is no longer possible; a possible execution is represented in green. Both algorithms can be implemented in linear time $O(m)$ by storing for each Eve-vertex the number of outgoing -edges which do not lead to the current reachability region.

saving a linear factor from the naive implementation. We now give more details and introduce the standard terminology.

Validity in ϕ. Fix a progress measure $\phi : V \to L$. Given an edge $e = v \xrightarrow{c} v'$ in G, we let $\phi(e) = \phi(v) \xrightarrow{c} \phi(v')$ and we say that e *is valid in* ϕ if $\phi(e)$ belongs to $\mathcal{L}$. Given a vertex $v \in V$, we say that it is *valid in* ϕ if either $v \in V_{\text{Eve}}$ and it has a valid outgoing edge, or $v \in V_{\text{Adam}}$ and all outgoing edges are valid. The following comes directly from the definition of Upd since $v \xrightarrow{c} v'$ is valid if and only if $\rho(\phi(v'), c) \leqslant \phi(v)$.

Lemma 4.3 (Rephrasing vertex-validity)

For all v in V, v is valid in ϕ if and only if $\text{Upd}(\phi)(v) \leqslant \phi(v)$.

A vertex or an edge which is not valid is *invalid*. Given a progress measure ϕ we let $\text{Inv}_\phi \subseteq V$ denote its set of invalid vertices and we let $\text{Cnt}_\phi : V_{\text{Eve}} \to \omega$ assign to each $v \in V_{\text{Eve}}$ its number of outgoing invalid edges.

Global value iteration. Recall that all points visited in a Kleene iteration are postfixpoints: we have $\phi_1 = \text{Upd}(\phi_\perp) \geqslant \phi_\perp$ since it is the minimal element and this inequality propagates to the whole sequence by induction thanks to monotonicity of Upd.

Given a subset $V' \subseteq V$ of vertices, we let $m_{V'}$ denote the number of edges adjacent to vertices in V', and we let $m_v = m_{\{v\}}$ be the number of edges adjacent to $v \in V$. In the statement below, the data is updated in place.

Lemma 4.4 (A step of global iteration)

Given a postfixpoint ϕ and assuming that Inv_ϕ and Cnt_ϕ are known, $\phi' = \mathrm{Upd}(\phi)$ as well as $\mathrm{Inv}_{\phi'}$ and $\mathrm{Cnt}_{\phi'}$ can be computed with runtime $O(m_{\mathrm{Inv}_\phi} t_{\mathcal{L}})$.

Proof. Since ϕ is a postfixpoint and by Lemma 4.3 a vertex v is invalid in ϕ if and only if $\mathrm{Upd}(\phi)(v) > \phi(v)$. Therefore ϕ' coincides with ϕ over ${}^{c}\mathrm{Inv}_\phi$.

We now compute (without erasing ϕ yet) ϕ' over Inv_ϕ which performs for each $v \in \mathrm{Inv}_\phi$ an access to ρ and a comparison for each outgoing edge, requiring runtime $\leqslant O(m_{\mathrm{Inv}_\phi} t_{\mathcal{L}})$.

There remains to compute $\mathrm{Cnt}_{\phi'}$ and $\mathrm{Inv}_{\phi'}$. First, for vertices $v \in \mathrm{Inv}_\phi \cap V_{\mathrm{Eve}}$, we compute $\mathrm{Cnt}_{\phi'}(v)$ from scratch in time $O(m_v t_{\mathcal{L}})$, and add them to $\mathrm{Inv}_{\phi'}$ if (and only if) $\mathrm{Cnt}_{\phi'}(v)$ is zero. Then we inspect all predecessors of vertices $v' \in \mathrm{Inv}_\phi$.

- If $e = v \xrightarrow{c} v'$ in G is such that $v \in V_{\mathrm{Adam}}$ and e is invalid in ϕ', then v is added to $\mathrm{Inv}_{\phi'}$.
- If $e = v \xrightarrow{c} v'$ in G is such that $v \in V_{\mathrm{Eve}} \setminus \mathrm{Inv}_\phi$ then we determine if e went from being valid to invalid in which case we decrement Cnt_ϕ. Again, if zero is reached v is added to $\mathrm{Inv}_{\phi'}$.

Both steps above can be done with runtime $O(t_{\mathcal{L}})$, which concludes: we may now replace values of ϕ over Inv_ϕ. □

The global value iteration computes the least fixpoint ψ by repeated applications of the lemma. Each vertex v is updated at most $|L|$ times and induces each time a runtime of $O(m_v t_{\mathcal{L}})$ which concludes with the announced complexity. Note that the space complexity for storing ϕ, Cnt and Inv is

$$O(n s_{\mathcal{L}} + n \log m),$$

where $s_{\mathcal{L}}$ is the size of the representation used for vertices of $\mathcal{L}$.

Local value iteration. The local value iteration algorithm is a non-deterministic variant which successively chooses an invalid vertex, updates it, and updates the data structured comprised of Inv and Cnt. This corresponds to the more usual variant (for instance the algorithms presented in [Jur00] and [BCD+11]), which is also a bit easier to implement. The non-deterministic aspect however makes it a bit harder to study.

Formally for each vertex v we define a corresponding *lifting operator* Lift_v *at* v over progress measures, by

$$\begin{array}{lcll} \mathrm{Lift}_v(\phi)(v) & = & \mathrm{Upd}(\phi)(v) & \\ \mathrm{Lift}_v(\phi)(v') & = & \phi(v') & \text{for } v' \neq v. \end{array}$$

In words the lifting operator at v updates v and leaves $v' \neq v$ untouched. It is clear that for each v the operator Lift_v is monotonous, and moreover ϕ is a fixpoint of Upd if and only if it is a fixpoint of Lift_v for all v.

Note that if ϕ is a postfixpoint of Upd then by monotonicity of ρ, so it $\mathrm{Lift}_v(\phi)$. Therefore the Kleene iteration can be generalised: starting from $\phi_\bot$, successive applications of Lift_v at invalid vertices preserve being a postfixpoint, and converges to the least fixpoint.

Again in the statement below, the data structures are updated in place.

Lemma 4.5 (A step of local iteration)

Given a postfixpoint ϕ and assuming that Inv_ϕ and Cnt_ϕ are known, for any $v \in \mathrm{Inv}_\phi$, $\phi' = \mathrm{Lift}_v(\phi)$ as well as $\mathrm{Inv}_{\phi'}$ and $\mathrm{Cnt}_{\phi'}$ can be computed with runtime $O(m_v t_{\mathcal{L}})$.

The proof is roughly the same as in the global case. It is a bit easier since we only care for one vertex.

Proof. We first compute in $O(m_v t_{\mathcal{L}})$ the value of ϕ' over v. If $v \in V_{\text{Eve}}$ then $\text{Cnt}_{\phi'}(v)$ is recomputed from scratch in time $O(m_v t_{\mathcal{L}})$ by inspecting validity of outgoing edges. We then update Inv and Cnt over predecessors of v.

- If $e = u \xrightarrow{c} v$ in G is invalid in ϕ' and $u \in V_{\text{Adam}}$ then u is added to Inv.
- If $e = u \xrightarrow{c} v$ in G is invalid in ϕ' and $u \in V_{\text{Eve}}$ then $\text{Cnt}(u)$ is decremented and if zero is reached, u is added to Inv. □

Again, each vertex is updated at most $|L|$ times, and each update induces time $O(m_v t_{\mathcal{L}})$. We conclude with the same worst-case complexity by summing over V:

runtime: $O(mt_{\mathcal{L}}|L|)$ space required: $O(ns_{\mathcal{L}} + n \log m)$.

3.2 Comparison of runtimes

The analysis we have provided above reports a worst-case runtime of $O(mt_{\mathcal{L}}|L|)$ for both the local and global iteration. In applications the linear dependency in $|L|$ is typically reached (for all executions of the local iteration, as well as for the global iteration) with very simple examples of constant size such as a vertex with a self-loop.

We now carry a more precise discussion comparing the runtimes of the two variants when the arena is fixed. We use it_{glob} to refer to the number of Kleene iterations of Upd, which is also the number of iterations of the global value iteration. Different executions of the local value iteration algorithm may lead to different numbers of iterations and different runtimes over the same arena.

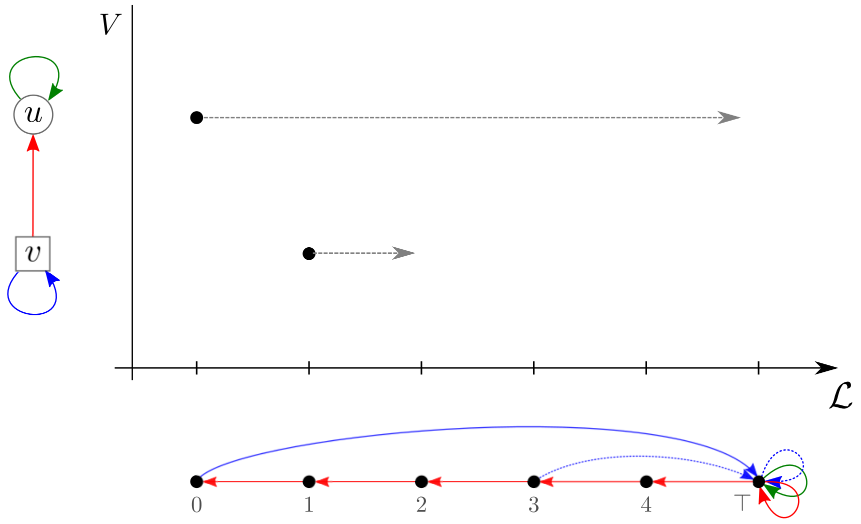


Figure 4.4: A game G and a completely monotonic graph $\mathcal{L}$ with colours $C = \{\text{green}, \text{blue}, \text{red}\}$; as always some edges which follow from composition (for instance the dotted ones) are not represented on $\mathcal{L}$. If $\phi(u) = 0$ and $\phi(v) = 1$, updating at u sets it to $\top$ and updating at v sets it to 2. Starting from the initial progress measure, the global iteration terminates in two steps, and so does the fastest local iteration (update u and then v). However, there is a local iteration (update v five times and then u) requiring six steps.

Lemma 4.6 (Optimality of the global iteration)

Any execution of the local iteration requires at least $\mathrm{it_{glob}}$ iterations.

We let ϕ_0 denote the initial progress measure.

Proof. Consider an execution of the local algorithm, and let ϕ_i be the progress measure obtained just after the i-th iteration. We show by induction that $\phi_i \leqslant \mathrm{Upd}^i(\phi_0)$, which implies the result. The equality holds for $i = 0$. Assume $\phi_i \leqslant \mathrm{Upd}^i(\phi_0)$ for some i. Then for some v, $\phi_{i+1} = \mathrm{Lift}_v(\phi_i) \leqslant \mathrm{Upd}(\phi_i)$ by definition of the lift operators and $\mathrm{Upd}(\phi_i) \leqslant \mathrm{Upd}^{i+1}(\phi_0)$ by induction hypothesis and monotonicity of Upd. □

Hence, in terms of number of iterations, there is no gain in applying the local iteration. Now performing a global update requires a number of operations which is proportionnal to the number of edges adjacent to invalid vertices, whereas the cost of a local update to v is proportionnal to the number of edges adjacent to v.

So there might still be a gain (of a multiplicative factor of at most m) in applying the local iteration rather than the global one. However, there is a risk (see Figure 4.4) of a poor choice in the way the updates are performed, which can lead to (greatly) suboptimal runtime. We now propose a simple way to perform the local iteration which avoids such bad scenarios.

Lemma 4.7 (A policy for local value iteration)

Consider a local iteration which always updates the invalid vertex $v' \in \mathrm{Inv}_\phi$ which was last updated. It performs $\leqslant n \cdot \mathrm{opt}$ iterations.

Proof. Let ϕ be a postfixpoint obtained during an execution of the policy under study, and $\mathrm{Inv}_\phi = \{v_0, \dots, v_r\}$ be its set of invalid vertices, ordered such that v_i was updated prior to v_j if $i < j$. For each $i \in [1, r+1]$, we let ϕ_i denote the progress measure obtained i steps after ϕ. Then ϕ_1 is obtained from ϕ by updating v_0, and moreover $v_1, v_2, \dots, v_r$ are invalid in ϕ_1 since we have for $i \in [1, r]$,

$$\mathrm{Upd}(\phi_1)(v_i) \geqslant \mathrm{Upd}(\phi)(v_i) > \phi(v_i) = \phi_1(v_i).$$

It follows that ϕ_2 is obtained from ϕ_1 by updating v_1, and by repeating the above argument in an inductive step for all $i \in [0, r]$, ϕ_{i+1} is obtained from ϕ_i by updating v_i (it is understood here that $\phi_0 = \phi$, and we apologize for the conflict in notations).

Now, for all $i \in [0, r]$, we have

$$\phi_{r+1}(v_i) \geqslant \phi_{i+1}(v_i) = \mathrm{Upd}(\phi_i)(v_i) \geqslant \mathrm{Upd}(\phi)(v_i),$$

and we conclude that $\phi_{r+1} \geqslant \mathrm{Upd}(\phi)$, since vertices not in I_ϕ satisfy $\mathrm{Upd}(\phi)(v) = \phi(v)$.

It follows by induction that the sequence $\phi_0, \phi_1, \dots$ of postfixpoints obtained along the execution satisfy that if i_j is given by $i_0 = 0$ and $i_{j+1} = i_j + |I_{\phi_{i_j}}| \leqslant i_j + n$ we have for all j

$$\phi_{i_j} \geqslant \mathrm{Upd}^j(\phi_0),$$

which yields the wanted result. □

Note that the proof applies to any "round-based" updating policy, where it is forbidden to update the same vertex twice if another vertex, which was already invalid at first, has not been updated since.

It is not hard to see that in such an execution, the runtime (in terms of total number of elementary operations) is better than that of the global iteration, but by at most a multiplicative factor of V. We do not expand on this for the sake of brevity.

Summing up the discussion,

(i) no local execution is much faster than the global iteration,

(ii) some local executions may be much worse,

(iii) "round-based" local executions are at least as good (but not much better thanks to (i)).

Finite monotonic graphs for parity games 5

Throughout this chapter we fix an even number d and study the parity objective over $C = [0, d]$ given by

$$\mathrm{Parity}_{[0,d]} = \{w \in [0, d]^\omega \mid \limsup w \text{ is even}\}.$$

We denote it Parity for short, and often simply say n-universal instead of (n, Parity)-universal.

We have obtained in Chapter 2 an n-universal monotonic graph of size $n^{d/2}$, by lexicographical combination of Co-Büchi (or Büchi) objectives. This corresponds to the signatures extracted by Walukiewicz [Wal96] from the proof of Emerson and Jutla [EJ91], and the induced value iteration algorithm is the one of Jurdziński [Jur00], with runtime $O(mn^{d/2})$.

Organisation of Chapter 5. The first section simplifies the quest for parity-universal graphs by establishing a bijection[1] between parity-saturated graphs and ordered trees. Therefore, parity-universal graphs of minimal size correspond to universal trees of minimal size, which are easier to handle. Our presentation relies on *relevant occurrences* which were introduced in [BJW02].

The second section presents the Jurdziński and Lazić's construction of a quasipolynomial universal tree [JL17], and Fijalkow's almost matching lower bound [Fij18].

1 From even graphs to ordered trees

Even cycles. We say that a finite graph satisfying Parity is an *even graph*. Being even for a finite graph is a property of its cycles, and we say that a cycle is *even* if its maximal priority is even. We fix a finite graph G over V of size $|V| = n$.

Lemma 5.1 (Cycles in even graphs)

It holds that G is even if and only if all cycles in G are even.

Proof. Assume that G is even. Let π_0 be a cycle in G. Repeating π_0 induces an infinite path $\pi = \pi_0^\omega$ in G with $\mathrm{col}(\pi) = \mathrm{col}(\pi_0)^\omega$. Then $\limsup \mathrm{col}(\pi) = \max \mathrm{col}(\pi_0)$ is even thus π_0 is even.

Conversely, assume G has only even cycles, and pick an infinite path π in G. Then there is a vertex v visited infinitely often by π, and then π may be decomposed into an infinite sequence

[1] This is a slight approximation: parity-saturated graphs may have $\xrightarrow{0}$-equivalent vertices which one should quotient to obtain an ordered tree (see below).

of cycles $\pi = \pi_0\pi_1\dots$ which all start and end in v. Then $\limsup \operatorname{col}(\pi) = \limsup_i \max \operatorname{col}(\pi_i)$, which is even since for all i, $\max \operatorname{col}(\pi_i)$ is even. □

Relevant occurrences. Consider a finite or infinite word of priorities $w = p_0 p_1 \dots \in [0, d]^{\leqslant\omega}$. We say that an occurrence i of the priority p_i is *relevant* in w if for all $j \leqslant i$ we have $p_j \leqslant p_i$. Below, a word w of priorities in which we have underlined the relevant occurrences:

$$w = \underline{2}01\underline{3}2\underline{3}1\underline{5}4\underline{5}123\underline{5}2222\dots.$$

For each *odd* priority $p \in \{1, 3, \dots, d-1\}$, we let $\operatorname{occ}_p(w) \in \omega + 1$ denote the (possibly infinite) number of relevant occurrences of p in w, and we define the *vector of odd relevant occurrences* of v by

$$\operatorname{occ}(w) = (\operatorname{occ}_{d-1}(w), \operatorname{occ}_{d-3}(w), \dots, \operatorname{occ}_1(w)) \in (\omega + 1)^{d/2}.$$

For the word w above, we have $\operatorname{occ}(w) = (3, 2, 0)$, if $d = 6$.

Occurrences are extended from words of priorities to paths in graphs simply by considering the colouration of the path. The following statement adds precision to Lemma 5.1 by a pumping argument.

Lemma 5.2 (Odd occurrences in even graphs)

It holds that G is even if and only if $\operatorname{occ}(\pi)$ is bounded for all paths π of G. Moreover in this case, all coordinates of $\operatorname{occ}(\pi)$ are bounded by $n - 1$.

Proof. If G does not satisfy even, then it has an odd cycle, say with maximal priority $p \in \{1, 3, \dots, d-1\}$. Repeating the odd cycle produces a path π with $\operatorname{occ}_p(\pi) = \omega$.

Conversely assume that G has a path $\pi = v_0 \xrightarrow{p_0} v_1 \xrightarrow{p_1} \dots$ such that $\operatorname{occ}_p(w) \geqslant n$ for some odd p, and let $i_1 < i_2 < \dots < i_n$ denote the first n relevant occurrences of p in π: $p_{i_j} = p$ for all j, and we have $p_i \leqslant p$ for all $i \leqslant i_n$. Then the $n + 1$ vertices $v_0, v_{i_1+1}, v_{i_2+1}, \dots, v_{i_n+1}$ in this order, all have a path with maximal priority p to the next one. There must be a repetition in this sequence, which induces an odd cycle, therefore G does not satisfy Even. □

Assuming G is even and ordering tuples of integers lexicographically, we may define a map

$$\begin{array}{rccl} \operatorname{occ}_G : & V & \to & \omega^{d/2} \\ & v & \mapsto & \operatorname{occ}_G(v) = \max\limits_{v \overset{w}{\rightsquigarrow} \text{ in } G} \operatorname{occ}(w) \end{array}$$

which takes values in $[0, n-1]^{d/2}$. It is actually easy to see that occ coincides with $\operatorname{val}_G^{\mathcal{L}_n}$ where $\mathcal{L}_n$ is the universal well-monotonic graph for parity games of size n introduced in Chapter 2 (this is also true with $\mathcal{L}_\alpha$ for any ordinal $\alpha \geqslant n$).

In particular, $\operatorname{occ}_G(V)$ is a finite subset of tuples of $\omega^{d/2}$ and we shall see such tuples as representing occurrences of odd priorities, and in particular denote them by $u = (u_{d-1}, u_{d-3}, \dots, u_3, u_1) \in \omega^{d/2}$. We let $h = d/2$. It is natural in this context to consider over ω^h an increasing sequence of $h + 1$ total preorders

$$\geqslant_1 \subseteq \geqslant_3 \subseteq \dots \subseteq \geqslant_{d-1} \subseteq \geqslant_{d+1}$$

defined by lexicographically comparing the truncations up to index p, formally

$$(u_{d-1}, u_{d-3}, \dots, u_1) \geqslant_p (u'_{d-1}, u'_{d-3}, \dots, u'_1) \iff (u_{d-1}, \dots, u_p) \geqslant_{\text{lex}} (u'_{d-1}, u'_{d-3}, \dots, u'_p).$$

As always, we use the standard notations relative to total preorders: $=_p$ denotes the equivalence relation associated with $\geqslant_p$, whose equivalence classes are strictly ordered by $>_p$. Note that $\geqslant_{d+1}$ is the full relation, it has one equivalence class. The order $\geqslant_1$ is antisymmetric; it is a total order which coincides with $\geqslant_{\text{lex}}$, and $=_1$ coincides with the equality over T.

Ordered trees. A (finite, ordered) *tree* of height $h \geqslant 0$ is a finite subset T of ω^h. In this context, we think of elements of a tree as representing occurrences of odd priorities and thus use the same notation as above $u = (u_{d-1}, u_{d-3}, \ldots, u_3, u_1) \in T$.

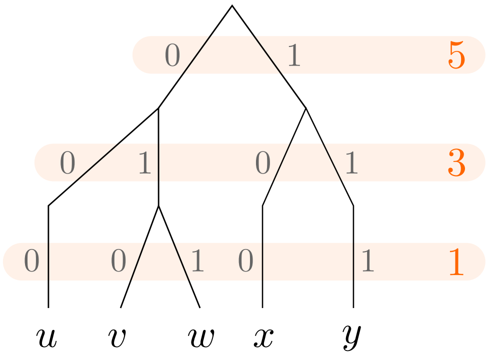


Figure 5.1: The tree $T = \{000, 010, 011, 100, 111\} = \{u, v, w, x, y\}$. It has height $h = 3$. The three levels, indexed with odd integers, are represented in orange. We have $u =_5 v =_5 w <_5 x =_5 y$, and $u <_3 v =_3 w <_3 x <_3 y$.

Note that there is a unique tree of height 0 which we call the *empty tree*, and that trees of height 1 are identified with sets of integers equipped with $\geqslant$.

Tree-morphisms and universality. What matters to us in a tree is the induced sequence of preorders. Given two trees T_1 and T_2 of height h, we say that a map $\phi : T_1 \to T_2$ is a *tree-morphism* if it preserves all preorders:

$$\forall p \in \{1, 3, \ldots, d-1\}, \forall u, u' \in T_1, \qquad u \geqslant_p u' \iff \phi(u) \geqslant_p \phi(u').$$

As for graphs, we say in this case that T_1 maps into T_2 or that T_2 embeds T_1. Note that such a morphism is necessarily injective: if $u > u'$ then $\phi(u) > \phi(u')$ since $>$ coincides with $>_1$.

Two trees are isomorphic if there is a morphism in both ways, which means intuitively that T_1 and T_2 are the same up to renaming. Morphisms therefore correspond to what is sometimes called "tree-pruning": T_1 maps into T_2 if and only if T_1 can be obtained (up to isomorphism) by removing elements of T_2.

A tree T of height h is n*-universal* if it embeds all trees of height h and size $\leqslant n$. We also say in this case that T is (h, h)-universal.

From trees to monotonic graphs. A tree T of height h induces a $[0, d]$-graph $\mathcal{L}_T$ over T given by[2]

$$u \xrightarrow{p} u' \text{ in } \mathcal{L}_T \qquad \iff \qquad \begin{array}{lll} p \text{ is even and} & v \geqslant_{p+1} v' & \text{or} \\ p \text{ is odd and} & v >_p v'. & \end{array}$$

Note that $u \xrightarrow{p} u'$ in $\mathcal{L}_T$ implies in general $u \geqslant_{p'} u'$ for odd $p' \geqslant p$.

[2] Again, note the similarity with Chapter 2.

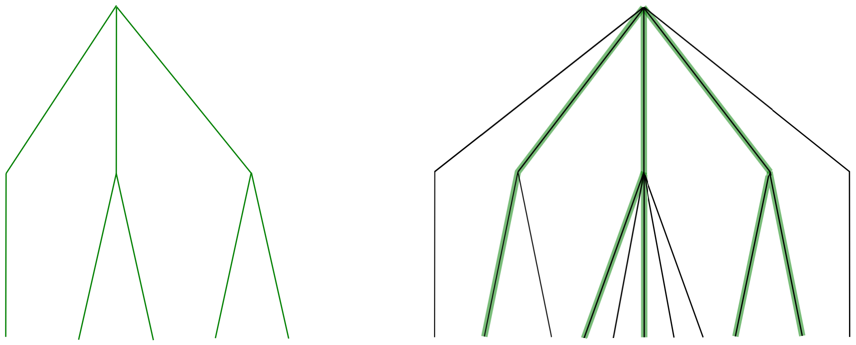


Figure 5.2: On the right, a tree of height 2 and size 11. This tree is $(2,5)$-universal, and is actually of minimal size. On the left, a tree of size 5 and one possible embedding in the universal tree.

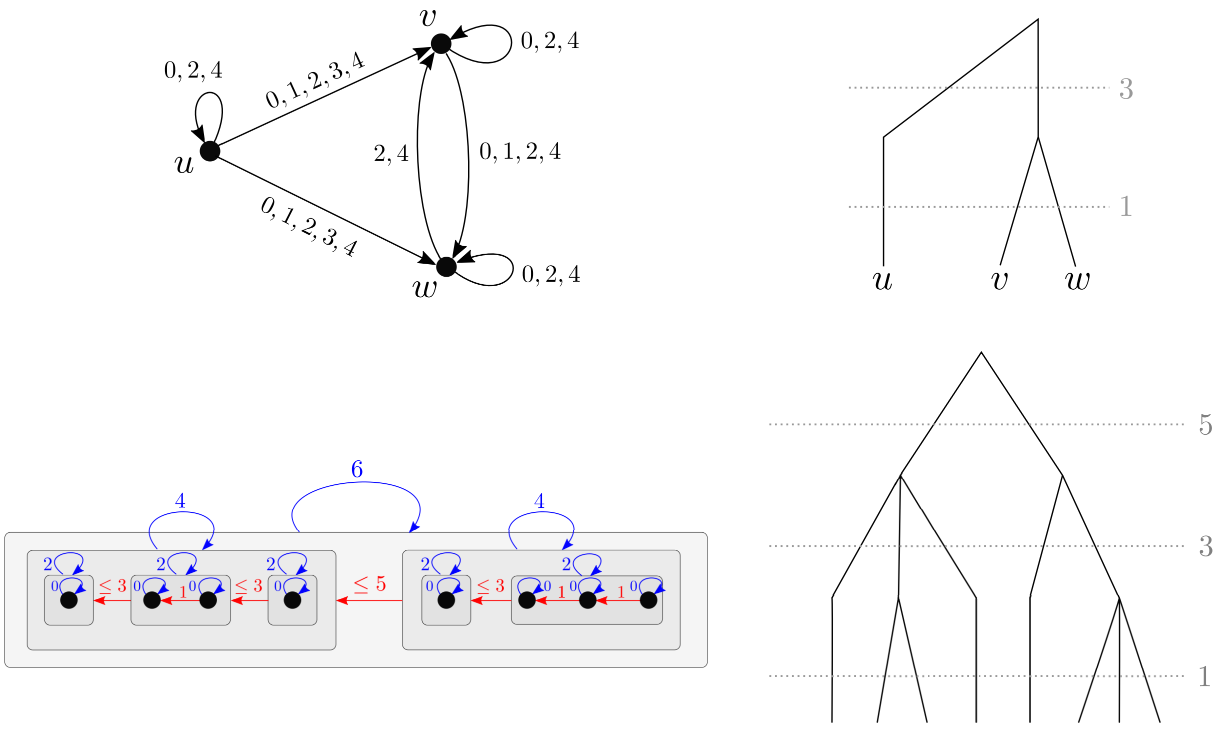


Figure 5.3: Two examples; each time the graph on the left correspond to the tree which is depicted on the right. For the bottom one, we use boxes to represent edges more efficiently.

Lemma 5.3 (Graphs of trees)

Let T be a tree of height $h = d/2$. Its graph $\mathcal{L}_T$ is an even Parity-saturated monotonic graph.

Proving that $\mathcal{L}_T$'s are saturated is not technically required but we believe it to be an important feature. Actually, it is not hard to see that (up to contracting vertices which have the same predecessors and successors) all Parity-saturated graphs are of this form.

Proof. We first show that $\mathcal{L}_T$ satisfies Parity. Let $\pi : u_0 \xrightarrow{p_0} u_1 \xrightarrow{p_1} \dots$ be an infinite path in $\mathcal{L}_T$ and assume by contradiction that $p = \limsup_i p_i$ is odd: for all $i \geqslant i_0 \in \omega$ we have $p_i \leqslant p$ and for infinitely many i's $p_i = p$. Then for all $i \geqslant i_0$ we have $u_i \geqslant_p u_i'$ and this inequality is strict for infinitely many i's, a contradiction.

We now show that adding any edge to $\mathcal{L}_T$ yields a graph which does not satisfy Parity. Let $e = u \xrightarrow{p} u'$ be an edge that does not appear in $\mathcal{L}_T$. If p is even, then $u' >_{p+1} u$, so the edge $u' \xrightarrow{p+1} u$ belongs to $\mathcal{L}_T$ hence $(\mathcal{L}_T)_e$ contains an odd cycle. If p is odd, then $u' \geqslant_p u$, which implies that the edge $u' \xrightarrow{p-1} u$ belongs to $\mathcal{L}_T$, and again $(\mathcal{L}_T)_e$ contains an odd cycle.

Finally, monotonicity of $\mathcal{L}_T$ with respect to $\leqslant_1$ follows simply from the inclusions $\geqslant_1 \subseteq \geqslant_p$ for each p; we prove it below for completeness and there are four similar cases.

- *Left composition.* Let $u, u', u'' \in T$ and $p \in [0, d]$ be such that $u \geqslant_1 u'$ and $u' \xrightarrow{p} u''$. If p is odd we have $u \geqslant_p u'$ and $u' >_p u''$ which yields $u >_p u''$ thus $u \xrightarrow{p} u''$. If p is even we have $u \geqslant_{p+1} u'$ and $u' \geqslant_{p+1} u''$ implying $u \geqslant_{p+1} u''$ and $u \xrightarrow{p} u''$.

- *Right composition.* Let $u, u', u'' \in T$ and $p \in [0, d]$ be such that $u \xrightarrow{p} u'$ and $u \geqslant_1 u''$. If p is odd we have $u >_p u'$ and $u' \geqslant_p u''$ which implies the wanted result. If p is even we have $u \geqslant_{p+1} u' \geqslant_{p+1} u''$ and again, the result follows.

□

Tree of relevant occurrences. We have seen that $\mathrm{occ}_G(V)$ defines a tree if and only if G is even, and that trees can be equipped with a monotonic even graph structure. We now close the cycle.

Lemma 5.4 (Parity-structuration)

Let G be an even graph and let $T = \mathrm{occ}_G(V)$. Then $\mathrm{occ}_G : V \to T$ defines a graph-morphism from G to $\mathcal{L}_T$.

This corresponds to a structuration result since $\mathcal{L}_T$ is even and smaller than G. This is really an explicit version of Theorem 3.1: $\mathcal{L}_T$ can be obtained from G by saturation (with all colours). Note however that there are other[3] saturations of G in general, which correspond to other trees.

Proof. Let $e = v_0 \xrightarrow{p_0} v_1$ be an edge in G and consider a path $\pi_1 : v_1 \xrightarrow{p_1} v_2 \xrightarrow{p_2} \dots$ such that $\mathrm{occ}(\pi_1) = \mathrm{occ}(v_1)$. Then $\pi_0 = e\pi_1$ is a path from v_0 in G therefore $\mathrm{occ}(v_0) \geqslant \mathrm{occ}(\pi_0)$. Any relevant occurrence of a priority $\geqslant p_0$ in π_1 is relevant in π_0. Therefore if p_0 is even then $\mathrm{occ}(\pi_0) \geqslant_{p_0+1} \mathrm{occ}(\pi_1)$, and if p_0 is odd then $\mathrm{occ}(\pi_0) >_{p_0} \mathrm{occ}(\pi_1)$ since 0 is an additional occurrence of p_0 in π_0. □

Rephrasing Parity-universality. To move from Parity-universality over graphs to tree-universality, there remains to see that the notions of morphisms coincide.

Lemma 5.5 (Morphisms of graphs and trees)

Let T_1 and T_2 be two trees of height $d/2$, and let $\phi : T_1 \to T_2$. Then ϕ is a tree-morphism if and only if it is a graph-morphism from $\mathcal{L}_{T_1}$ to $\mathcal{L}_{T_2}$.

[3] To be more precise, any prefixpoint of $\mathrm{Upd}_G^{\mathcal{L}_\omega}$ defines such a structuration of G, where $\mathcal{L}_\omega$ is the well-monotonic graph from Chapter 2 (which also corresponds to $\mathcal{L}_{\omega^{d/2}}$ if $\omega^{d/2}$ is seen as an (infinite) tree). In this regards $\mathrm{occ}_G(V)$ is minimal; it is the smallest prefixpoint.

Proof. Recall that by definition, $u >_p u'$ is the negation of $u \leqslant_p u'$. Hence we have

$$\begin{aligned}
\phi \text{ is a tree-morphism} \quad &\iff \quad \forall u, u' \in T_1 \text{ and } p \text{ odd}, \ (u \geqslant_p u' \iff \phi(u) \geqslant_p \phi(u')) \\
&\iff \quad \forall u, u' \in T_1 \begin{cases} \forall p \text{ even} & (u \geqslant_{p+1} u' \implies \phi(u) \geqslant_{p+1} \phi(u')) \\ \forall p \text{ odd} & (u >_p u' \implies \phi(u) >_p \phi(u')) \end{cases} \\
&\iff \quad \forall u, u' \in T_1 \text{ and } p, u \xrightarrow{p} u' \in \mathcal{L}_{T_1} \implies \phi(u') \xrightarrow{p} \phi(u') \in \mathcal{L}_{T_2} \\
&\iff \quad \phi \text{ is a graph-morphism.}
\end{aligned}$$

□

We may finally present our main result in this section: Parity-universality is equivalent to tree-universality.

Theorem 5.1 (Equivalence between universalities)

If G is $(\text{Parity}_{[0,d]}, n)$-universal then $\text{occ}_G(V)$ is $(d/2, n)$-universal as a tree. Conversely, if T is (h, n)-universal then $\mathcal{L}_T$ is $(\text{Parity}_{[0,d]}, n)$-universal.

Proof. Assume that G is n-universal and let T be a tree of height $d/2$ and size $\leqslant n$. Then $\mathcal{L}_T$ has a morphism into G, which gives a morphism in $\mathcal{L}_{\text{occ}_G(V)}$ by Lemma 5.4. By Lemma 5.5 this gives a tree-morphism from T to $\text{occ}_G(V)$ which concludes.

Conversely, assume that T is n-universal and let G be an even graph of size $\leqslant n$ over V. Then $\text{occ}_G(V)$ is a tree of size $\leqslant n$ therefore it maps into T. Lemma 5.5 transfers this as a graph morphism from $\mathcal{L}_{\text{occ}_G(V)}$ to $\mathcal{L}_T$, which gives by Lemma 5.4 and composition on the left a morphism from G to $\mathcal{L}_T$. □

This shifts our focus from constructing n-universal (monotonic) graphs to constructing n-universal trees.

2 Universal trees and their size

Throughout this section, we let $d = 2h$, and still use odd integers as indexes for tree elements, keeping in mind the connexion to parity games. We will show almost matching quasipolynomial upper and lower bounds on the size of universal trees.

2.1 A quasipolynomial construction

We now present the very elegant construction of [JL17], which relies on encoding tree-elements with bitstrings.

Theorem 5.2 (A quasipolynomial upper-bound)

Let $n, h \geqslant 1$. There exists a (h, n)-universal tree of size

$$2n\binom{\lfloor \log n \rfloor + h - 1}{h - 1}.$$

It is convenient to present the construction when n is of the form $n_k = 2^{k+1} - 1$, and thus $\lfloor \log(n_k) \rfloor = k$. Note that we have $n_0 = 1$ and $n_{k+1} = 2n_k + 1$. We will define for all k a (n_k, h)-universal tree $T_{h,k}$ of size

$$n_k \binom{h+k-1}{h-1},$$

which implies the theorem in general by rounding n up to the next integer of the form n_k.

Infix ordering of bitstrings. We call *bitstrings* the set $\{0,1\}^*$ of finite words of *bits*. Consider the infix ordering $\geqslant$ over bitstrings $s \in \{0,1\}^*$ which is the linear order defined by the two equations

$$1s > \varepsilon > 0s'$$

and

$$ss' \geqslant ss'' \iff s' \geqslant s''$$

for all bitstrings $s, s', s'' \in \{0,1\}^*$.

Algorithmically, two bitstrings can be compared in linear time by scanning from left to right to determine their longest common prefix and then comparing the remainder using the first above equation.

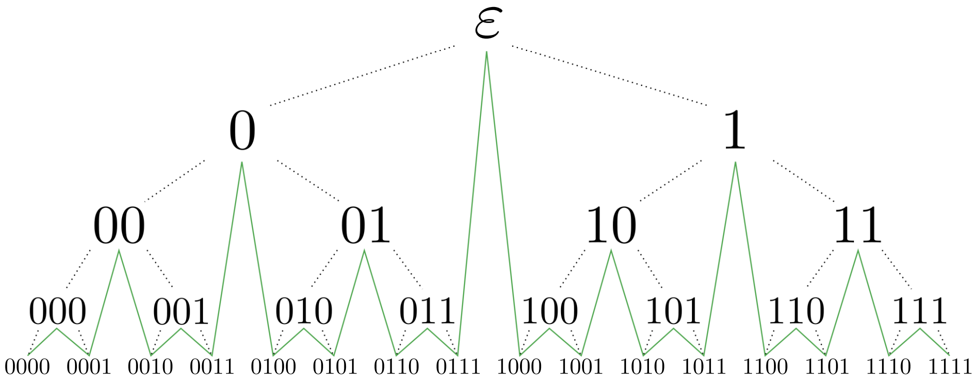


Figure 5.4: The infix ordering, read from left to right and symbolised with the green line, over bitstrings of length $\leqslant 4$.

We will only consider bitstrings of length $\leqslant k$ and for technical reasons we need to fix an integer encoding of bitstrings $\iota : \{0,1\}^{\leqslant k} \to \omega$, satisfying

$$s \geqslant s' \iff \iota(s) \geqslant \iota(s').$$

Using ι is required only because we have defined trees via integers and not arbitrary linear orders. It can generally be ignored below and should really be thought as an inclusion of strings in ω which respects $\leqslant$.

The construction. We define $T_{h,k}$ to be (recall that $h = d/2$)

$$T_{h,k} = \{(\iota(s_{d-1}), \iota(s_{d-3}), \ldots, \iota(s_1)) \in \omega^h \mid s_{d-1}s_{d-3}\ldots s_1 \in \{0,1\}^{\leqslant k}\}.$$

In words, elements of $T_{h,k}$ are obtained by splitting a bitstring s of length $\leqslant k$ into a concatenation $s_{d-1}s_{d-3}\ldots s_1$ of h smaller bitstrings. Elements of $T_{h,k}$ can then be compared via $\geqslant_p$ for odd priorities p by comparing the first few smaller bitstrings, where bitstring are compared via $\succcurlyeq$ and the overall comparison is performed lexicographically.

Elements of $T_{h,k}$ are described with a bitstring of length $\leqslant k$ and a split of the integer k into h non-negative summands $k_1, k_2, \dots, k_h$ such that $k_1 + \cdots + k_h = k$. There are $n_k = 2^{k+1} - 1$ such bitstrings, and $\binom{h+k-1}{h-1}$ such splits, hence the announced size.

Theorem 5.3 (Universality of $T_{h,k}$)

For all $h, k \in \mathbb{N}$, the tree $T_{h,k}$ is (h, n_k)-universal.

Stated differently, given a tree of size $\leqslant n_k = 2^{k+1} - 1$ and height h we aim to label its branching directions with bitstrings in an order-preserving fashion, and so that all branches use at most a total of k bits. This is achieved top-down via the recursive algorithm illustrated in Figure 5.5.

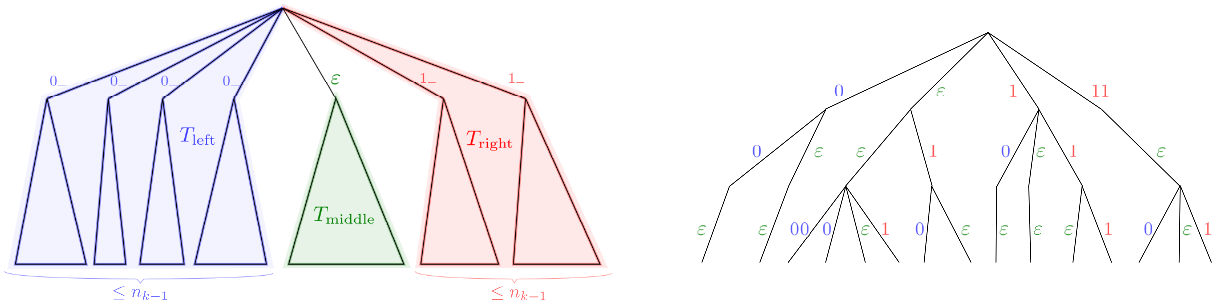


Figure 5.5: On the left, illustration of a recursive algorithm producing the wanted labelling. First identify $T_{\text{left}}, T_{\text{middle}}$ and T_{right} such that $|T_{\text{left}}|$ and $|T_{\text{right}}|$ are $\leqslant n_{k-1}$ (there may be several options). Then define the first bit of the first string of elements in T_{left} to be 0, in T_{right} to be 1, and the first string to be ε over $\mathcal{T}_{\text{middle}}$. Finally, recurse on the three trees. On the right, an example on a tree of size 15. Such a labelling corresponds to a tree-morphism into $T_{h,k}$.

Formal details are provided below.

Proof of Theorem 5.3. We prove the result by induction on h and k. For $h = 0$, $T_{h,k}$ is the empty tree which is trivially (h, n)-universal for all n : the empty tree is the only tree of height 0, and it embeds in itself, via the empty map. For $k = 0$ we have $n_k = 1$ and $T_{h,k}$ is the unique (up to isomorphism) tree of height h and size 1, and it embeds all trees of height h and size $\leqslant 1$. We now fix $h, k > 0$ and assume the result known for smaller values.

Let T be a tree of height h and of size $\leqslant n_k$. Let $i_1 < i_2 < \cdots < i_\ell$ be such that

$$T = i_1 T_{i_1} \cup i_2 T_{i_2} \cup \cdots \cup i_\ell T_{i_\ell},$$

where the T_{i_j}'s are (necessarily disjoint) non-empty trees of height $h-1$, and let $t_j = |T_{i_j}|$. Since $t_1 + t_2 + \cdots + t_\ell \leqslant n_k = 2n_{k-1} + 1$, there is an index j_0 such that both $t_{\text{left}} = t_1 + \cdots + t_{j_0-1}$ and $t_{\text{right}} = t_{j_0+1} + \cdots + t_\ell$ are $\leqslant n_{k-1}$.

We now let (see Figure 5.5)

$$T_{\text{left}} = i_1 T_{i_1} \cup \cdots \cup i_{j_0-1} T_{i_{j_0}-1} \quad \text{and} \quad T_{\text{right}} = i_{j_0+1} T_{i_0+1} \cup \cdots \cup i_\ell T_{i_\ell}.$$

Since T_{left} and T_{right} have size $\leqslant n_{k-1}$ we obtain by induction tree-morphisms ϕ_{left} and ϕ_{right} respectively from T_{left} and T_{right} to $T_{h,k-1}$. Likewise the induction hypothesis provides us with a tree-morphism ϕ_{middle} from $T_{\text{middle}} = T_{i_{j_0}}$ to $T_{h-1,k}$. We let $\phi : T \to T_{h,k}$ be given by

$$\phi(u) = \begin{cases} (\iota(\varepsilon), s_{d-3}, \dots, s_1) & \text{where } (s_{d-3}, \dots, s_1) = \phi_{\text{middle}}(u') & \text{if } u = i_{j_0}u' \in i_{j_0}T_{\text{middle}} \\ (\iota(0\iota^{-1}(s_{d-1})), s_{d-3}, \dots, s_1) & \text{where } (s_{d-1}, \dots, s_1) = \phi_{\text{left}}(u) & \text{if } u \in T_{\text{left}} \\ (\iota(1\iota^{-1}(s_{d-1})), s_{d-3}, \dots, s_1) & \text{where } (s_{d-1}, \dots, s_1) = \phi_{\text{right}}(u) & \text{if } u \in T_{\text{right}}. \end{cases}$$

We now prove that ϕ defines a tree-morphism. Let $u_1, u_2 \in T$ and let p be an odd priority, we aim to prove that

$$u_1 \geqslant_p u_2 \iff \phi(u_1) \geqslant_p \phi(u_2).$$

We let $\alpha_1, \alpha_2 \in \{\text{left, middle, right}\}$ be the respective types of u_1 and u_2, where the type of u is defined to be left if $u \in T_{\text{left}}$, middle if $u \in i_{j_0}T_{\text{middle}}$ and right otherwise. Types are of course linearly ordered by right $\geqslant$ middle $\geqslant$ left. Observe that for $p = d - 1$, we have on one hand,

$$u_1 \geqslant_p u_2 \iff \alpha_1 \geqslant_p \alpha_2.$$

On the other hand, since ι is increasing and since $0r > \varepsilon > 1r'$ whatever the bitstrings r, r' we also have,

$$\phi(u_1) \geqslant_p \phi(u_2) \iff \alpha_1 \geqslant_p \alpha_2.$$

Now assume $p < d - 1$. If u_1 and u_2 have different types, then

$$u_1 \geqslant_p u_2 \iff u_1 \geqslant_{d-1} u_2,$$

and we conclude as above.

Thus we assume that u_1 and u_2 have the same type α. If $\alpha =$ left then we have

$$u_1 \geqslant_p u_2 \iff \phi_{\text{left}}(u_1) \geqslant_p \phi_{\text{left}}(u_2) \iff \phi(u_1) \geqslant_p \phi(u_2),$$

where the equivalence on the right follows from monotonicity of ι and the second property defining $\geqslant$. The proof is analogous for $\alpha =$ right.

Finally if $\alpha =$ middle then we have $u_1 = i_{j_0}u_1'$ and $u_2 = i_{j_0}u_2'$ and

$$u_1 \geqslant_p u_2 \iff u_1' \geqslant_p u_2' \iff \phi_{\text{middle}}(u_1') \geqslant_p \phi_{\text{middle}}(u_2') \iff \phi(u_1) \geqslant_p \phi(u_2). \qquad \square$$

Note that the above procedure labels a given tree of size n_k in polynomial time. This does not really matter however for running value iterations in $\mathcal{L}_{T_{h,k}}$ which require to perform comparisons, and computations of min-predecessors.

Efficient navigation in $T_{h,k}$. Fix h and k to be positive integers. We assume that an element $u = (\iota(s_{d_1}), \iota(s_{d-3}), \dots, \iota(s_1)) \in T_{h,k}$ is represented by the string[4]

$$s(u) = s_{d-1}\#s_{d-3}\#\dots\#s_1 \in \{0, 1, \#\}^{\leqslant k+h-1}.$$

We say that 0, 1 and # are *characters*, which we order by $1 > \# > 0$.

We show that the required operations can be computed in linear time $O(k + h)$. Locally to this section, given $u \in T_{h,k}$ and an odd integer $p \in \{1, 3, \dots, d + 1\}$ we use $u_{\min,=_p}$ and $u_{\min,>_p}$ to denote respectively the smallest elements in $T_{h,k} \cup \{\top\}$ which are $=_p u$ and $>_p u$. Note that $u_{\min,=_p} \in T_{h,k}$ while $u_{\min,>_p} \in T_{h,k} \cup \{\top\}$.

[4] This is not always optimal (see discussion below), but more convenient.

Lemma 5.6 (Navigation in $T_{h,k}$)

- One can compare $u_1, u_2 \in T_{h,k}$ (with respect to $\geqslant_1$) from their string representation in time $O(h+k)$.
- One can compute $s(u_{\min,=_p})$ from $s(u)$ in time $O(h+k)$.
- One can compute $s(u_{\min,>_p})$ from $s(u)$ (or determine that $u_{\min,>_p}$ is $\top$) in time $O(h+k)$.

Proof.

- This is done simply by scanning $s_1 = s(u_1)$ and $s_2 = s(u_2)$ until finding the first character that differs. If either s_1 or s_2 is a strict prefix of the other then it is smaller. Otherwise, we have $s_1 = sb_1s_1'$ and $s_2 = sb_1s_2'$, and in this case $u_1 \geqslant u_2$ if and only if $b_1 \geqslant b_2$.
- We let $s(u) = s_{d-1}\#s_{d-3}\#\dots\#s_1$ and let $x = |s_{d-1}| + \dots + |s_p|$. Then we have
$$s(u_{\min,=_p}) = s_{d-1}\#\dots\#s_p\#0^{k-x}\#^{\lfloor p/2\rfloor - 1}.$$
- Using the notations from the previous item, we distinguish two cases.
  - If $x < k$, then we have
  $$s(u_{\min,>_p}) = s_{d-1}\#\dots\#s_p10^y\#^{\lfloor p/2\rfloor},$$
  where $y = \max(0, k-x-1)$.
  - If $x = k$, then we let p' be the smallest odd integer $\geqslant p$ such that $s_{p'} \neq \varepsilon$.
    - If $s_{p'}$ has a 0, we write $s_{p'} = s_{p'}'01^y$ with $y \geqslant 0$ and then we have
    $$s(u_{\min,>_p}) = s_{d-1}\#\dots\#s_{p'+2}\#s_{p'}'\#\dots\#0^{y+1}\#^{\lfloor p/2\rfloor}.$$
    - Otherwise, $s_{p'} = 1^y$ for some $y > 0$.
      - If $p' = d-1$ then $u_{\min,>_p} = \top$, and
      - If $p' < d-1$ then
      $$s(u_{\min,>_p}) = s_{d-1}\#\dots\#s_{p'-2}10^{y-1}\#^{\lfloor p/2\rfloor+1}.$$ □

This is sufficient for our algorithmic needs in terms of navigating $\mathcal{L}_{T_{h,k}}$.

Lemma 5.7 (Navigation in $\mathcal{L}_{T_{h,k}}$)

One can compare elements of $\mathcal{L}_{T_{h,k}}$ and compute min-predecessors in time $O(h+k)$.

Max-successors and min-predecessors in $\mathcal{L}_T$ for a given T are represented in Figure 5.6.

Proof. Linear-time comparison is already presented in Lemma 5.6. Then for all $u \in T_{h,k}$ and $p \in [0,d]$ we have
$$\rho(u,p) = \begin{cases} u_{\min,=_{p+1}} & \text{if } p \text{ is even} \\ u_{\min,>_p} & \text{if } p \text{ is odd} \end{cases}$$
and the wanted result follows via Lemma 5.6. □

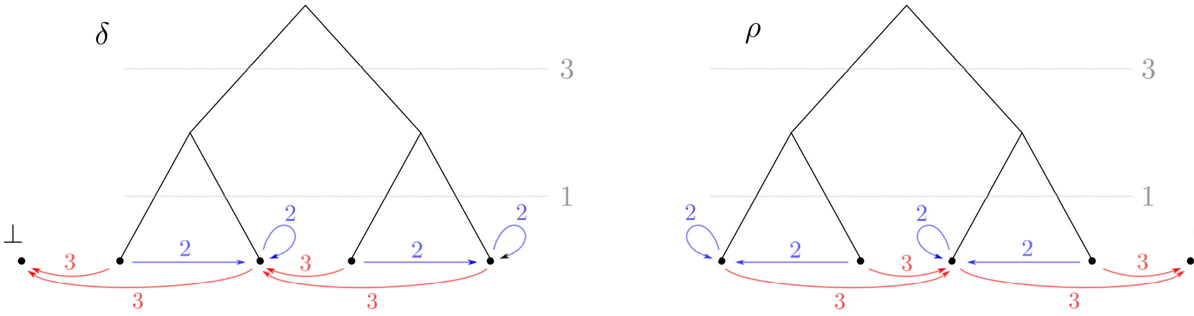


Figure 5.6: Depiction of max-successors (on the left) and min-predecessors (on the right) for priorities 2 and 3 in $\mathcal{L}_T$ where $T = \{00, 01, 10, 11\}$.

Summing up, performing the (local or global) value iteration algorithms in the monotonic graph $\mathcal{L} = \mathcal{L}_{T_{h,k}}$ can be done with quasilinear runtime and quasilinear space requirement, and more precisely

$$\text{runtime: } O\left(m\underbrace{(d+\lfloor\log n\rfloor)}_{t_{\mathcal{L}}}\underbrace{n\binom{\lfloor\log n\rfloor + d/2-1}{d/2-1}}_{|L|}\right) \qquad \text{space req.: } O(n\underbrace{(d+\log n)}_{s_{\mathcal{L}}}+n\log m).$$

If d is polylogarithmic in n, the expression on the left is polynomial in n.

Note that elements of $T_{h,k}$ can also be encoded over $O(\log d \log n)$ bits by writing, for each non-empty string, the string followed by an $\log d$ integer identifying the height. Lemmas 5.6 and 5.7 can easily be adapted to this encoding, and we obtain the alternative expression $O(\log d \log n)$, replacing $O(d + \log n)$ for both $t_{\mathcal{L}}$ and $s_{\mathcal{L}}$. This is the point of view adopted in [JL17]. It is better when d is superlogarithmic, and worse when d is sublogarithmic.

In any case, $|L|$ remains by far the dominating term. We refer to [JL17] (Lemma 6 and Theorem 7 therein) for precise expressions in different $\{\text{super}, \text{sub}, \varepsilon\} \cdot$ logarithmic regimes. To date, this is the most efficient algorithm for solving parity games in the worst case; similar bounds are given in [FJS+17] for their algorithm, which also apply to the one of [CJK+17].

2.2 An almost matching lower bound

We now present the lower bound of [Fij18] which matches the upper bound up to a factor of $2n$.

Theorem 5.4 (Lower bound on size of universal trees)

Any (h, n)-universal tree has at size at least

$$\binom{\lfloor\log n\rfloor + h - 1}{h-1}.$$

We separate the combinatorial argument with the analysis of the recursion it leads to.

Combinatorial argument. Fix a tree T of height $h \geqslant 1$, and let T' be the tree of height $h-1$ obtained by removing the lowest level of T, formally

$$T' = \{u' \in \omega^{h-1} \mid \exists r \in \omega, u'r \in T\}.$$

Moreover for each $u' \in T'$ we let $s_{u'} > 0$ be the number of such r, that is $s_{u'} = \{r \in \omega \mid u'r \in T\}$. Note that $|T| = \sum_{u' \in T'} s_{u'}$. Given $n' \in \omega$, we let $T'_{n'}$ be the tree of height $h-1$ obtained from T' by restricting to its elements u' satisfying $s_{u'} \geqslant n'$. (For instance, $T'_1 = T'$.)

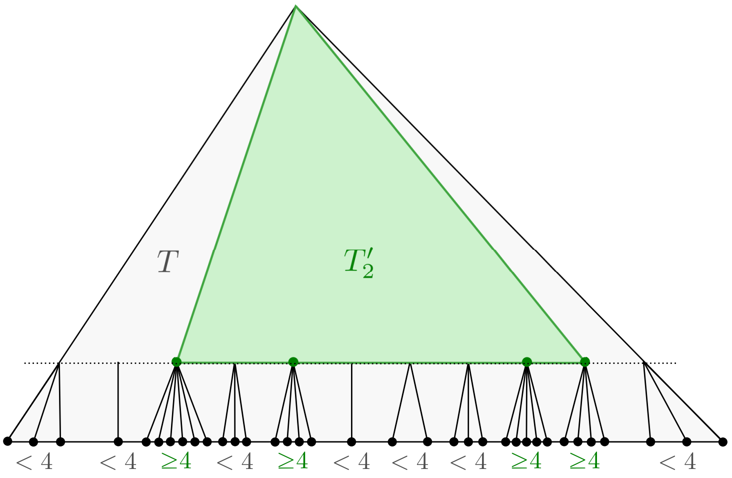


Figure 5.7: In black a tree T and in green the tree T'_4.

The lower bound relies on the following result.

Lemma 5.8 (Combinatorial argument)

Let $n, h \geqslant 1$ and assume that T is (h, n)-universal. Then for all $n' \in [1, n]$ it holds that $T'_{n'}$ is $(h-1, \lfloor n/n' \rfloor)$-universal.

Proof. Let $n' \in [0, k]$ and let $\tilde{T}'$ be a tree of height $h-1$ with $\leqslant \lfloor n/n' \rfloor$ leaves. We let $\tilde{T}$ be obtained from $\tilde{T}'$ by appending n' children at each leaf, formally

$$\tilde{T} = \tilde{T}' \times |n'|.$$

Note that $\tilde{T}$ has size $n'|\tilde{T}'| \leqslant n$ and height h therefore there is a tree-morphism ϕ from $\tilde{T}$ to T.

Now, for each $u' \in \tilde{T}'$, it holds that $u'0, u'1, \dots, u'(n'-1) \in \tilde{T}$ are pairwise distinct and $=_3$-equivalent (that is, they share all but the last coordinates), hence so must be their image by ϕ in T. Stated differently let $\phi'(u) \in \mathbb{Z}^{h-1}$ defined over $u \in \tilde{T}'$ by restricting to all but the last coordinate of $\phi'(u \cdot 0)$, formally

$$\phi'(u) = (\phi(u'0)_{d-1}, \dots, \phi(u'0)_3).$$

It satisfies that there exist n' pairwise distinct integers $r \in \omega$ such that $\phi'(u')r \in T$ and thus $s_{\phi(u')} \geqslant n'$. Hence, ϕ' maps $\tilde{T}'$ in $T'_{n''}$ and the fact that ϕ' defines a tree-morphism follows directly from the fact that ϕ does. □

Now observe that we have

$$|T| = \sum_{u' \in T'} s_{u'} = \sum_{n'=1}^{n} \sum_{u' \in T'} \mathbb{1}_{s_{u'} \geqslant n'} = \sum_{n'=1}^{n} |T'_{n'}|,$$

where $\mathbb{1}_x$ equals 1 if x holds and 0 otherwise. We let $g(h,n)$ denote the size of the smallest (h,n)-universal tree.

Analysis of the recursion. The above equation together with the lemma translate into the recursion

$$g(h,n) \geqslant \textstyle\sum_{n'=1}^{n} g(h-1, \lfloor n/n' \rfloor),$$

with base case $g(0,n) = 1$ for all n. By restricting to powers of 2 and setting $n = 2^k$, we obtain

$$g(h, 2^k) = G(h,k) \geqslant \textstyle\sum_{k'=1}^{k} G(h-1, k'). \tag{$*$}$$

Developing the recursion h times yields

$$G(h, k_h) \geqslant \textstyle\sum_{k_{h-1}=1}^{k_h} G(h-1, k_{h-1}) = \sum_{k_{h-1}=1}^{k_h} \sum_{k_{h-2}=1}^{k_{h-1}} G_{h-2,k_{h-2}} = \cdots = \sum_{k_h \geqslant k_{h-1} \geqslant \ldots k_0} 1.$$

Therefore, $G(h,k)$ is at least the number of non-decreasing sequences in $[1,k]^h$, which is

$$\binom{h+k-1}{h-1}.$$

This yields the theorem by truncating n down to $2^{\lfloor \log n \rfloor}$.

6 Finite monotonic graphs for mean-payoff games

Reminders from Chapter 2. We now consider the threshold mean-payoff objective over finite $\mathbb{Z}$-arenas, given by

$$\mathrm{MP}^{\leqslant 0} = \{t_0 t_1 \cdots \in \mathbb{Z}^\omega \mid \limsup_n \frac{1}{n} \sum_{i=0}^{n-1} t_i \leqslant 0\}.$$

We also recall the definition of the energy valuation

$$\mathrm{Energy}^+(t_0 t_1 \dots) = \sup_n \sum_{i=0}^{n-1} t_i \in [0, \infty],$$

which we proved (Lemma 2.14) to be intimately linked to the threshold mean-payoff objective over finite graphs since we have

$$\mathrm{MP}(v) \leqslant 0 \quad \iff \quad \mathrm{Energy}^+(v) < \infty \quad \iff \quad \mathrm{Energy}^+(v) \leqslant (n-1)N$$

over $[-N, N]$-graphs of size n. We say that a finite graph is *bounded* if it satisfies the threshold mean-payoff objective, which is therefore equivalent to having $\mathrm{Energy}^+_G \leqslant (n-1)N$ or having only non-positive cycles.

Recall that we denote $\mathrm{MP}^{\leqslant 0}_A$ for the restriction of the objective to weights in $A \subseteq \mathbb{Z}$. A finite $(\mathrm{MP}^{\leqslant 0}_A, n)$-universal graph is simply a bounded graph which embeds all bounded A-graphs of size $\leqslant n$. When $A = [-N, N]$, we use $\mathrm{MP}^{\leqslant 0}_N$ for simplicity.

Recall the well-monotonic graph $\mathcal{L}$ over $L = \omega$ introduced for energy games given by

$$\ell \xrightarrow{t} \ell' \text{ in } \mathcal{L} \quad \iff \quad t \leqslant r - r'.$$

Given $B \subseteq \omega$ we let $\mathcal{L}_B$ denote the restriction of $\mathcal{L}$ to B.

We have seen that all paths in $\mathcal{L}$ have bounded profile, and therefore non-positive mean-payoff, hence $\mathcal{L}$ satisfies $\mathrm{MP}^{\leqslant 0}$. We have also seen that Energy^+_G defines a graph morphism from any graph G to $\mathcal{L}^\top$. Therefore if G is finite and bounded, Energy^+_G embeds G in $\mathcal{L}_{[0,(n-1)N]}$, which was stated as Corollary 2.3: $\mathcal{L}_{[0,(n-1)N]}$ is $(\mathrm{MP}^{\leqslant 0}_N, n)$-universal.

The associated value iteration algorithms have complexity

$$\text{runtime: } O(n \underbrace{mN}_{|\mathcal{L}|} \underbrace{\log(nN)}_{t_{\mathcal{L}}}) \qquad \text{space required: } O(n \underbrace{(\log(nN))}_{\log|L|} + n \log m),$$

and the local variant corresponds exactly to the BCDGR algorithm [BCD+11].

Organisation of Chapter 6. The first section studies upper and lower bounds for $(\mathrm{MP}^{\leqslant 0}, n)$-universal (monotonic) graphs. We start by showing a $N^{1-1/n}$ lower bound, and complement this result with a subtler construction with matching size, up to a polynomial factor. Using our construction rather than the simple one above is meaningful only in the regime where $n^n = O(N)$, in which we dispose of better algorithms (see Chapter 10).

It is easy to see that one may restrict in general to arenas with at most n^2 edges. Therefore if the absolute value N of the largest weight is much larger (for instance exponential) than n, the set of weights A used on a given arena if a very sparse subset of $[-N, N]$. As an important example consider $A = \{-(-n)^p \mid p \in [0, d]\}$ in which case $\mathrm{MP}_A^{\leqslant 0}$ coincides with Parity over arenas of size $\leqslant n$ (see preliminaries). In this case we know from the previous chapter that there is a universal graph of quasipolynomial size $O(n^{\log d})$, which is much smaller than $nN = n^{d+1}$.

This raises the following question: can we find general bounds on the size of $(\mathrm{MP}_A^{\leqslant 0}, n)$-graphs which are parameterised on the cardinality k of A? This question is tackled in the second section, where we show that there is always a universal (monotonic) graph of size $O(n^k)$ but (for some well-chosen A) it may be that the smallest universal graphs have size $\Omega(n^{k-1})$.

1 Universal graphs for $A = [-N, N]$

We show in this section upper and lower bounds for $(\mathrm{MP}_N^{\leqslant 0}, n)$-universal monotonic graphs (or equivalently, universal graphs, by the structuration results of Chapter 3). We start with a very useful lemma.

Zero-cycles. Note that in a bounded graph the two suprema in

$$\mathrm{Energy}_G^+(v) = \sup_{v \overset{t_0 t_1 \dots}{\rightsquigarrow}} \sup_{n \in \mathbb{N}} \sum_{i=0}^{n-1} t_i$$

are reached since the value is finite. Therefore for each v there exists a finite path from v whose sum coincides with the value of v, and we say that such a path is *tight*.

We say that a cycle whose sum is zero is a *zero-cycle*.

Lemma 6.1 (Zero-cycles and values)

Let $\pi : v_0 \xrightarrow{t_0} v_1 \xrightarrow{t_1} \dots \xrightarrow{t_{k-1}} v_0$ be a cycle of length k in a bounded graph. Then for all $i \in [0, k-1]$, $\mathrm{Energy}^+(v_i) - \mathrm{Energy}^+(v_0)$ coincides with the sum $t_0 + \dots + t_{i-1}$ of $\pi_{<_i}$.

Proof. Let π' be a tight path from v_i. Then $\pi_{<i}\pi'$ defines a path from v_0 therefore

$$\mathrm{Energy}^+(v_0) \geqslant \mathrm{Energy}^+(v_i) + (t_0 + \dots + t_{i-1}).$$

Applying the same result to the cycle $v_i \xrightarrow{t_i} \dots \xrightarrow{t_{k-1}} v_0 \xrightarrow{t_0} \dots \xrightarrow{t_{i-1}} v_i$ yields

$$\begin{aligned}\mathrm{Energy}^+(v_i) &\geqslant \mathrm{Energy}^+(v_0) + (t_i + \dots t_{k-1}) \\ &= \mathrm{Energy}^+(v_0) - (t_0 + \dots + t_{i-1}),\end{aligned}$$

the wanted converse inequality. □

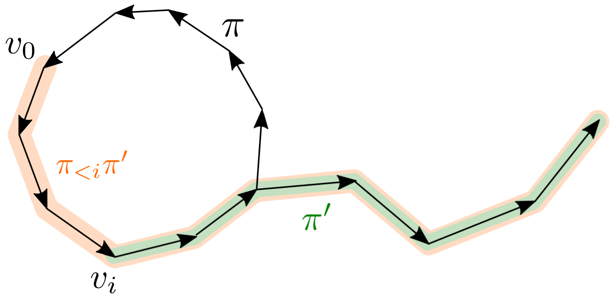


Figure 6.1: The paths in the proof of Lemma 6.1.

We will rather use a consequence of the lemma which is the following.

Corollary 6.1 (Preservation of differences)

Let G and G' be two bounded graphs and let ϕ be a morphism from G to G'. If there is a zero-cycle visiting both v_1 and v_2 then

$$\text{Energy}^+_G(v_1) - \text{Energy}^+_G(v_2) = \text{Energy}^+_{G'}(\phi(v_1)) - \text{Energy}^+_{G'}(\phi(v_2)).$$

This result simply follows from the lemma and the fact that paths and their sums (and in particular, zero-cycles) are preserved.

Lower bound. We start with a lower bound.

Theorem 6.1 (Lower bound for $A = [-N, N]$)

Any $(\text{MP}^{\leqslant 0}_N, n)$-universal graph has size at least $N^{1-1/n}$.

Proof. Let $H = [0, N]^{n-1}$, and for each $h = h_1 \dots h_{n-1} \in H$ consider the $[-N, N]$-graph G_h over $[0, n-1]$ given by exactly the edges

$$i \xrightarrow{h_i} i - 1 \qquad \text{and} \qquad i \xrightarrow{-h_i} i - 1.$$

for $i \in [1, n-1]$. See Figure 6.2.

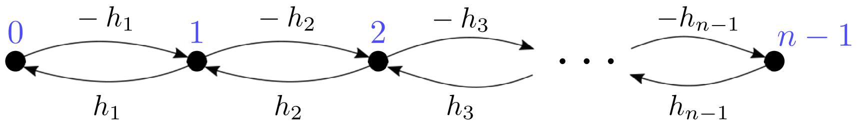


Figure 6.2: The graph G_h.

Note that $\text{Energy}^+_{G_h}(i) = h_1 + \dots + h_i$ for all $i \in [0, n-1]$, and in particular G_h is bounded. Note moreover that

$$0 \xrightarrow{-h_1} 1 \xrightarrow{-h_2} \dots \xrightarrow{-h_{n-1}} n-1 \xrightarrow{h_{n-1}} n-2 \xrightarrow{h_{n-2}} \dots \xrightarrow{h_1} 0$$

defines a zero-cycle in G_h which visits all vertices.

Let G be a $(\mathrm{MP}_N^{\leqslant 0}, n)$-universal graph over V and fix for each $h \in H$ a morphism ϕ_h from G_h to G. Consider the map

$$\begin{array}{rccl} f: & H & \to & V^n \\ & h & \mapsto & (\phi_h(0), \dots, \phi_h(n-1)). \end{array}$$

By Corollary 6.1 we have for all h and for all $i \in [1, n-1]$ that

$$\mathrm{Energy}^+_G(\phi_h(i)) - \mathrm{Energy}^+_G(\phi_h(i-1)) = \mathrm{Energy}^+_{G_h}(i) - \mathrm{Energy}^+_{G_h}(i-1) = h_i$$

and therefore f is injective. We conclude that $|V|^n \geqslant |H| = (N+1)^{n-1}$ which yields the announced lower bound. □

Therefore there is no hope in finding a monotonic graph which is $(\mathrm{MP}_N^{\leqslant 0}, n)$-universal and implies a value iteration algorithm which is (strongly) sublinear in N.

Upper bound. We have already seen a $(\mathrm{MP}_N^{\leqslant 0}, n)$-universal monotonic graph of size $O(nN)$, which yields the value iteration algorithm of [BCD+11]. We now describe a more subtle construction.

Theorem 6.2 (A slightly more succint construction)

There exists a $(\mathrm{MP}_N^{\leqslant 0}, n)$-universal monotonic graph of size

$$2\Big((n-1)N - \big[((n-1)N)^{1/n} - 1\big]^n\Big) \leqslant 2n \cdot (nN)^{1-1/n}.$$

Note that the expression on the left is $\leqslant 2nN$. The majoration on the right gives an improvement over $O(nN)$ when N is much greater than n, for instance if $n^n = o(N)$. We do not know of an analysis of the left-hand side which gives a bound better than $O(nN)$ for smaller values of N. We now give a high-level explanation of the construction.

We have seen that any bounded graph maps into the restriction $\mathcal{L}_{S(G)}$ of $\mathcal{L}$, where $S(G)$ is the set of Energy$^+$-values of vertices in G. Note that $\mathcal{L}_{S(G)}$ is smaller than G and therefore we may restrict our attention to graphs of the form $\mathcal{L}_A$ where $A \subseteq \omega$. Indeed, a graph of this form $\mathcal{L}_B$ is $(\mathrm{MP}_N^{\leqslant 0}, n)$-universal if and only if it embeds all $\mathcal{L}_A$'s where $A = S(G)$ for some G of size $\leqslant n$: such graphs must be embedded, and it is sufficient to embed them by composition.

For the first construction $\mathcal{L}_{[0,(n-1)N]}$ we have used the fact that $A = S(G)$ in this case is always *included* in $B = [0, (n-1)N]$, and therefore the identity maps $\mathcal{L}_A$ into $\mathcal{L}_B$. What we exploit now is that an embedding of $\mathcal{L}_A$ into $\mathcal{L}_B$ is not necessarily an inclusion of A into B; it suffices that $A + p \subseteq B$ for some $p \in \mathbb{Z}$. This will allow us to remove some unnecessary values from $B = [0, (n-1)N]$ while remaining universal. As a small drawback, we have to double the range to $[0, 2(n-1)N]$.

We fix $b = ((n-1)N)^{1/n}$, and write integers $a \in [0, 2(n-1)N)$ in basis b, hence using $n+1$ digits written $a[i] \in [0, b)$, formally

$$a = \sum_{i=0}^{n} a[i] b^i = \sum_{i=0}^{n} a[i]((n-1)N)^{i/n}.$$

Note that since $a \in [0, 2(n-1)N)$ the $(n+1)$-th digit is either 0 or 1.

We let B be the set of integers in $[0, 2(n-1)N)$ which have at least one zero digit among the first n digits in this decomposition. Note that B is obtained by removing $2(b-1)^n$ elements from $[0, 2(n-1)N]$ which explains the left-hand side in the theorem.

Lemma 6.2 (Main ingredient for Theorem 6.2)

Let $A = \{d_0, d_1, \ldots, d_{n-1}\} \subseteq \omega$ be such that $0 = d_0 < d_1 < \cdots < d_{n-1} \leqslant (n-1)N$. There exists $p \in \omega$ such that $A + p \subseteq B$.

It is not hard to see that the lemma implies the theorem: given a bounded $[-N, N]$-graph G of size n, its set $S(G)$ of Energy$^+$-values satisfies the hypotheses of the lemma, and therefore $\mathcal{L}_{S(G)}$ embeds into $\mathcal{L}_B$.

Proof of the lemma. We choose p of the form $\sum_{i=0}^{n-1} a_i b^i$ for $a_i \in [0, b)$. Note that the $(n+1)^{\text{th}}$ digit is 0, or equivalently $p < (n-1)N$. Let us write $p_i = p + d_i$, for $i \in [0, n-1]$. We need to choose p such that for every $i \in [0, n-1]$ we have $p_i \in B$. Note that for all $i \in [0, n-1]$ it holds that $d_i \in [0, (n-1)N)$ so whatever the choice of $p \in [0, (n-1)N]$ we have $p_i \in [0, 2(n-1)N)$.

We show how to choose $a_0, a_1, \ldots, a_{n-1}$ in order to ensure that $p_0, p_1, \ldots, p_{n-1} \in B$, that is, each have at least a zero digit among the first n ones. More precisely, we show by induction on $k \in [0, n-1]$ that there exist $a_0, \ldots, a_k \in [0, b)$ such that for any choice of $a_{k+1}, \ldots, a_{n-1} \in [0, b)$ and for all $i \in [0, k]$, the i-th digit $p_i[i]$ of p_i is 0. For $k = 0$, we let $a_0 = 0$, which yields $p_0[0] = 0$ independently of the values of $a_1, \ldots, a_{n-1}$.

Let $a_0, \ldots, a_{k-1}$ be such that for any choice of $a_k, \ldots, a_{n-1}$ and for any $i \in [0, k-1]$ we have $p_i[i] = 0$. Let $a_k \in [0, b)$ be the unique value such that $\left(\sum_{i=0}^{k} a_i b^i + d_k\right)[k] = 0$. Let $a_{k+1}, \ldots, a_{n-1} \in [0, b)$. By induction hypothesis for any $i \in [0, k)$ we have $p_i[i] = 0$. Now

$$p_k[k] = (p + d_k)[k] = \left(\sum_{i=0}^{n-1} a_i b^i + d_k\right)[k] = \left(\sum_{i=0}^{k} a_i b^i + d_k\right)[k] + \left(b^{k+1} \sum_{i=k+1}^{n-1} a_i b^{i-k}\right)[k] = 0,$$

since both terms are zero. □

2 Universal graphs parameterised by $k = |A|$

We now fix a subset of weights $A \subseteq \mathbb{Z}$ and we let $k = |A|$. This section is devoted to finding upper and lower bounds over the size of $(\mathrm{MP}_A^{\leqslant 0}, n)$-universal graphs parameterised by k.

Upper bound. We start with an easy construction which requires a slight variation on Lemma 2.14. We say that a path is *simple* if it does not contain any cycle.

Lemma 6.3 (Existence of tights paths which are simple)

In a bounded graph all vertices have simple tight paths.

Note that a simple path has length $\leqslant n-1$ therefore this gives another (similar) proof of $(i) \implies (iv)$ in Lemma 2.14.

Proof. Let $v_0 \in V$ and let π be a tight path from v_0 of minimal size. If π is not simple, it is of the form $\pi = \pi_0 \pi_1 \pi_2$ where π_1 is a cycle. Thus π_1 has non-positive sum therefore $\pi_0 \pi_2$ is a tight path from v_0 which is smaller than π, a contradiction. □

Let B be the set of non-negative sums of $w \in A^{\leqslant n-1}$. Note that $|B|$ is bounded in general by $(n-1)^k$ since the sum of a word is invariant under permutation of its letters. Now we know that values of a bounded graphs coincide with sums of simple paths and therefore belong to B, which yields the following result.

Theorem 6.3 (Upper bound paramterised by k)

The monotonic graph $\mathcal{L}_B$ is $(\mathrm{MP}_A^{\leqslant 0}, n)$-universal and has size $\leqslant (n-1)^k$.

It is not immediate however how to compute min-predecessors in $\mathcal{L}_B$ in general, therefore some more work is required for turning this construction into a value iteration when given a set of weights A.

Lower bound. We now present a last lower bound result. Fix n and k, assume that $k-1$ divides $n-1$ and let $T = \{1 + n + n^2 + \cdots + n^{k-2}\}$ and consider

$$A = \{1, n, n^2, \ldots, n^{k-2}, -\frac{n-1}{k-1} T\}.$$

Note that A has cardinality k. We match the above upper bound up to a linear factor for this particular A.

Theorem 6.4 (Lower bound for A of small size)

Any $(\mathrm{MP}_A^{\leqslant 0}, n)$-universal graph has size $\Omega(n^{k-1})$.

The proof is similar to that of the lower bound in the first section (Theorem 6.1): we construct a family of graphs which have a zero-cycle and achieve many different Energy$^+$-values, which implies an injective map thanks to Corollary 6.1.

Proof. We let $q = \frac{n-1}{k-1}$ and let[1]

$$C = \left\{c_0 \ldots c_{k-2} \in \omega^{k-1} \mid \textstyle\sum_i c_i = q\right\}$$

be the set of sequences of $k-1$ non-negative integers who sum to q. Note that C has cardinality

$$|C| = \binom{q+k-1}{k-1}.$$

We let $H = C^{k-1}$ be the set of square matrices with columns in C, whose elements we denote by $h = (h_{i,j})_{i,j \in [0,k-2]}$ with for all j, $\sum_i h_{i,j} = q$. We fix $h \in H$. Given $(i_0, j_0) \in [0, k-2]^2$ we let s_{i_0,j_0} denote the partial sum of the matrix up to (and excluding) (i_0, j_0) when it is read row by row, formally

$$s_{i_0,j_0} = \sum_{(i,j) <_{\mathrm{lex}} (i_0,j_0)} h_{i,j}.$$

We now define a graph G_h (see Figure 6.3) over $[0, n-1]$ comprised of a unique cycle and given by the edges $0 \xrightarrow{-qT} n-1$ and

$$r+1 \xrightarrow{n^{j_0}} r \qquad \text{where } (i_0, j_0) = \max\{(i,j) \mid s_{i,j} \leqslant r\}.$$

[1] We apologize for the notation clash. Here C is of course not a set of colours but a set of columns.

In words, G_h is constructed from $h \in H$ by reading the matrix row by row, each time adding $h_{i,j}$-many n^j-edges pointing to the left, and closing the cycle with a $-qT$-edge. For convenience we also define $s_{k-1,0} = n - 1$.

$$h = \begin{bmatrix} 1 & 2 \\ 3 & 2 \end{bmatrix}$$

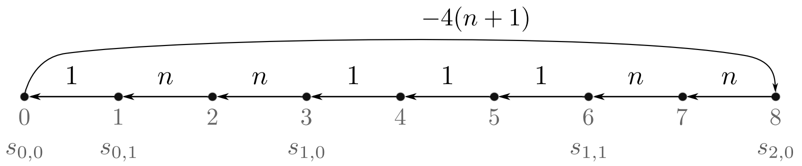


Figure 6.3: Depiction of G_h when h is the matrix on the left. Here, we have $k - 1 = 2, n - 1 = 8$ and $q = 4$; both columns sum to q.

Since all columns in h sum to q the cycle in G_h has sum 0 thus G_h is bounded and moreover by Lemma 6.1 we have for all $i \in [0, k-2]$ that

$$\text{Energy}^+_{G_h}(s_{i+1,0}) - \text{Energy}^+_{G_h}(s_{i,0}) = \sum_{j=0}^{k-2} h_{i,j} n^j.$$

Note that since the $h_{i,j}$'s are $\leqslant q \leqslant n - 1 < n$, one may recover the i-th row from the above right-hand side: they are the digits in the decomposition in basis n.

Let G be a $(\text{MP}^{\leqslant 0}_A, n)$-universal graph over V and fix a morphism ϕ_h from G_h to G for each $h \in H$. Then we have by Corollary 6.1 that for all $h \in H$ and $i \in [0, k-2]$,

$$\text{Energy}^+_G(\phi_h(s_{i+1,0})) - \text{Energy}^+_G(\phi_h(s_{i+1,0})) = \sum_{i=0}^{k-2} h_{i,j} n^j$$

and therefore

$$\begin{array}{rccl} f: & H & \to & V^{k-1} \\ & h & \mapsto & (\phi_h(s_{0,0}), \phi_h(s_{1,0}), \ldots, \phi_h(s_{k-2,0}) \end{array}$$

is injective since each of the $h_{i,j}$'s can be recovered from $f(h)$.

Hence we have $|V| \geqslant |H|^{1/(k-1)} = |C|$ which rewrites as

$$|C| = \binom{\frac{n-1}{k-1} + k - 1}{k-1} \geqslant \left(\frac{\frac{n-1}{k-1} + k - 1}{k-1} \right)^{k-1} \geqslant \left(\frac{n-1}{(k-1)^2} \right)^{k-1}$$

and implies $|V| = \Omega(n^{k-1})$ for k constant. □

Finite monotonic graphs for mean-payoff parity games 7

We fix an even integer d and a non-negative integer N, in this chapter the set of colours is $C = [0,d] \times [-N,N]$. We denote words by $w = (p_0,t_0)(p_1,t_1)\dots$, and use $w_{\text{prio}} = p_0 p_1 \dots$ and $w_{\text{wei}} = t_0 t_1 \dots$ to denote the projections of w over each coordinate, which are of different nature.

The mean-payoff parity objective is given by

$$W = \{w \in C^\omega \mid w_{\text{prio}} \in \text{Parity}_{[0,d]} \text{ or } w_{\text{wei}} \in \text{MP}_N^{\leqslant 0}\}$$

We refer to the introduction for a survey of the literature relative to mean-payoff parity games. We note that $(0,0) \in C$ is weakly neutral, and that positionality of W over finite arenas follows from the concavity of $\text{Parity}_{[0,d]}$ and $\text{MP}_N^{\leqslant 0}$, in the sense of Kopczyński [Kop06].

Our approach versus [DJL18]. In this chapter, we reobtain the main result of Daviaud, Jurdziński and Lazić [DJL18]: a value iteration algorithm for mean-payoff parity games with (pseudo-quasipolynomial) runtime $O(mnNS_{n,d}t)$, where $S_{n,d}$ is the size of a universal tree and t is the runtime for computing min-predecessors in the corresponding graph. There is a technical caveat here: the construction requires taking into account inner nodes of the tree, which is not a serious issue, but we no longer have a closed expression for $S_{n,d}$.

Our approach however differs from that of [DJL18] in that we present the algorithm using a universal monotonic graph. There are slight technical differences between the two constructions, most notably our graph is of the form $T \times [0,nN] \cup \{\top\}$, whereas theirs rather looks like $T \times ([0,nN] \cup \{\top\})$.

More importantly, our approach revolves around the study of graphs satisfying W, whereas theirs relies on strategy decompositions. This allows to significantly factor the argumentation, since our strategy decomposition (for Eve) is contained directly in the construction: we implicitly consider positional strategies σ described by morphisms $G_\sigma \to \mathcal{L}$. This avoids a tedious[1] additional definition, and non-trivial proofs[2] for equivalence between progress measures and strategy decompositions. At the same time, our notions roughly coincide, in particular our proofs of soundness (Lemma 1 in [DJL18] versus our Lemma 7.2) are somewhat similar.

But the difference lies not only in the presentation: their argument for existence of strategy decompositions follows the recursive template laid out by Zielonka [Zie98] (which is usual for mean-payoff parity games), and in particular it *involves the opponent*. Therefore in addition to the (already difficult) inductive argument, the framework requires understanding the structure of Adam

[1] Their definition uses 9 different cases, and introduces a number of new symbols, see page 6 in [DJL18].

[2] Pages 13-16 in [DJL18] establish the equivalence between progress measures and strategy decompositions.

strategies[3]. In particular, membership in coNP is derived, whereas it is not at all implied by our method. This is to be compared to our proof of universality (Lemmas 7.3 and 7.4) which is self-contained, combinatorial and might be of independent interest.

In the first section, we introduce our monotonic graph and prove that it satisfies W. In the second section, we prove its n-universality.

In [AFGL+21], we took a completely different approach by using deterministic separating automata. This leads to a much more direct 1 page proof, which moreover has the advantage of using separating automata for parity and mean-payoff games as black boxes. Two drawbacks: it is not clear how to obtain a value iteration algorithm (the construction is not naturally monotonous), or a universality proof (we believe that there is value in designing templates for such proofs).

1 Constructing a monotonic graph satisfying W

Trees and their preorders. We need a slight variation on the definition of trees presented in Chapter 5. Hopefully the notion of tree universality is robust and easily adapts, we give some details below. In this chapter, a (rooted, ordered) *tree* of height h is a finite subset T of $\omega^{\leqslant h}$. Stated differently, trees are now comprised of tuples of potentially different sizes. This is the point of view which was adopted in [JL17] and makes more sense when parity games are vertex-coloured; it is also used in [DJL18].

The lexicographical order is defined over ω^* as usually: $u \leqslant_{\text{lex}} u'$ when u is a prefix of u'. We still let $h = d/2$, index elements of $\omega^{d/2}$ with odd integers from $d-1$ to 1, and define the increasing sequence of preorders

$$\geqslant_1 \subseteq \geqslant_3 \subseteq \cdots \subseteq \geqslant_{d+1}$$

over $\omega^{d/2}$ by first (potentially) truncating up to index p and comparing the two obtained tuples lexicographically. Note that as previously $\geqslant_1$ coincides with the lexicographical order (no truncation is performed).

Tree morphisms are those maps which preserve all preorders, and still correspond to tree-pruning. Note that a tree $T \subseteq \omega^{\leqslant h}$ can be *padded* into a tree $\text{pad}(T) \subseteq \omega^h$ by adding zeros at the end of tuples of length $< h$. It is easy to see that $|\text{pad}(T)| \leqslant |T|$ and that T_1 embeds into T_2 if and only if $\text{pad}(T_1)$ embeds into $\text{pad}(T_2)$.

This allows to reduce to the definition of Chapter 5, and in particular the tree comprised of all prefixes of $T_{h,k}$ embeds (with respect to the new notion of morphisms) all trees of size $\leqslant n = 2^k$ and height h, it has quasipolynomial size as in [JL17] and is efficiently navigable.

Given an element $u = (u_{d-1}, u_{d-3}, \dots, u_p)$ in a tree of height $d/2$ we let $p_u = p$ denote its last index, which is always odd by definition.

The construction. We fix n and let $B = (n-1)N$. We introduce a notation which is often convenient: given a priority p, we let

$$\lceil p \rceil^{\text{odd}} = \begin{cases} p & \text{if } p \text{ is odd} \\ p+1 & \text{if } p \text{ is even,} \end{cases}$$

be obtained by rounding up to the nearest odd number.

[3] Understanding Adam's strategies corresponds to Section 2.3 (pages 7-9) in [DJL18], adds 10 slightly different cases, and another hard proof.

We now let $\mathcal{L}$ be the graph over $\omega^{\leqslant d/2} \times [0,B]$ given by

$$(u,r) \xrightarrow{(p,t)} (u',r') \qquad \iff \qquad \text{either} \quad \begin{array}{ll} (1a): & p+1 \geqslant p_u \text{ and } p \text{ even and } u \geqslant_{p+1} u' \\ (1b): & p \geqslant p_u \text{ and } p \text{ odd and } u >_p u' \\ (2a): & \lceil p \rceil^{\text{odd}} < p_u \text{ and } u > u' \\ (2b): & \lceil p \rceil^{\text{odd}} < p_u \text{ and } u = u' \text{ and } t \leqslant r - r'. \end{array}$$

We say that an edge is of type $(1a)$, $(1b)$, $(2a)$ or $(2b)$ if the corresponding condition is fulfilled.

Intuitively this combines the constructions of the two previous chapters: the global structure of the graph responds to large priorities and is inherited from the preorders over $\omega^{\leqslant d/2}$, whereas small priorities are "absorbed" and weights are read instead.

Elements in $L = \omega^{\leqslant d/2} \times [0,B]$ are well-ordered lexicographically.

Lemma 7.1

The graph $\mathcal{L}$ is well-monotonic.

Proof. We have to prove left and right composition in $\mathcal{L}$. Right composition is direct: in each case the condition over u's and $r's$ composes with the lexicographical order when p and t are fixed. Left composition is much more tedious since p_u changes from one edge to the other (see below) and therefore we have many similar cases to examine. We let $(u,r) \geqslant (u',r') \xrightarrow{(p,t)} (u'',r'')$ in $\mathcal{L}$, we let $e' = (u',r') \xrightarrow{(p,t)} (u'',r'')$ and aim to show that $e = (u,r) \xrightarrow{(p,t)} (u'',r'')$ belongs to $\mathcal{L}$.

- If $u = u'$. Then $r \geqslant r'$ and in all cases e has the same type as e'.
- If $u > u'$ and $p_u = p_{u'}$. Then again in all cases e has the same type as e'.
- If $u > u'$ and $p_u > p_{u'}$.
  - If $\lceil p \rceil^{\text{odd}} \geqslant p_u > p_{u'}$. Then e' has type $(1a)$ or $(1b)$ and we have $u \geqslant_{\lceil p \rceil^{\text{odd}}} u'$ therefore e has the same type as e'.
  - If $p_u > \lceil p \rceil^{\text{odd}} \geqslant p_{u'}$ then again e' has type $(1a)$ or $(1b)$ and we now have $u >_{\lceil p \rceil^{\text{odd}}} u'$ therefore $u > u''$ and e is of type $(2a)$.
  - If $p_u > p_{u'} > \lceil p \rceil^{\text{odd}}$ then $u > u' \geqslant u''$ therefore e is of type $(2a)$.
- If $u > u'$ and $p_{u'} > p_u$.
  - If $\lceil p \rceil^{\text{odd}} \geqslant p_{u'} > p_u$ then e' has type $(1a)$ or $(1b)$ and $u \geqslant_{\lceil p \rceil^{\text{odd}}} u'$ therefore e has the type of e'.
  - If $p_{u'} > \lceil p \rceil^{\text{odd}} \geqslant p_u$ then e' has type $(2a)$ or $(2b)$ therefore $u' \geqslant u''$ and we have $u >_{\lceil p \rceil^{\text{odd}}} u'$ thus e has type $(1a)$ or $(1b)$ according to the parity of p.
  - If $p_{u'} > p_u > \lceil p \rceil^{\text{odd}}$ then e' has type $(2a)$ and $(2b)$ and e has type $(2a)$ since we have $u > u' \geqslant u''$. □

Note that in every cases if $(u,r) \xrightarrow{(p,t)} (u',r')$ belongs to $\mathcal{L}$ then it holds that $u \geqslant_{\lceil p \rceil^{\text{odd}}} u'$, and therefore by inclusion of the preorders $u \geqslant_{p'} u'$ if p' is an odd priority $\geqslant p$. This property is crucial in the proof below.

Lemma 7.2

It holds that $\mathcal{L}$ satisfies W.

Proof. Consider an infinite path

$$\pi : (u_0, r_0) \xrightarrow{(p_0,t_0)} (u_1, r_1) \xrightarrow{(p_1,t_1)} \dots$$

in $\mathcal{L}$, assume that $p = \limsup_i p_i$ is odd and let $i_0 \in \omega$ be such that for all $i \geqslant i_0$ we have $p_i \leqslant p$. It follows that for all $i \geqslant i_0$ we have $u_i \geqslant_p u_{i+1}$. Now by well-foundedness of $>_p$ it must be that for some $i_1 \geqslant i_0$ all u_i's for $i \geqslant i_1$ are $=_p$-equivalent.

Consider $i_2 \geqslant i_1$ such that $p_{i_2} = p$ which is odd. The corresponding edge $e = (u_{i_2}, r_{i_2}) \xrightarrow{(p,t_{i_2})} (u_{i_2+1}, r_{i_2+1})$ cannot be of type $(1b)$ since $u_{i_2} =_p u_{i_2+1}$, therefore it is of type $(2a)$ or $(2b)$ and thus $\lceil p \rceil^{\text{odd}} = p < p_{u_{i_2}}$. But observe that a tuple u with $p_{u_{i_2}}$ is alone in its $=_p$-equivalence class: if $u' =_p u$ then $u' = u$.

Therefore we have for all $i \geqslant i_1$ that u_i is identical to u_{i_2}, and we denote it by u. Since moreover they have priority $\leqslant p < p_u$, all edges in

$$\pi_{\geqslant i_1} : (u, r_{i_1}) \xrightarrow{(p_{i_1},t_{i_1})} (u, r_{i_1+1}) \xrightarrow{(p_{i_1+1},t_{i_1+1})} \dots$$

are of type $(2b)$, and therefore for all $i \geqslant i_1$ we have

$$t_i \leqslant r_i - r_{i+1}.$$

Therefore we have a telescoping sum, and for all $k \geqslant i_1$ it holds that

$$\textstyle\sum_{i=i_1}^{k-1} t_i \leqslant r_i - r_k \leqslant r_i,$$

which implies that $r_0 r_1 \dots$ is bounded, and therefore π has non-positive mean-payoff. □

Given a tree $T \subseteq \omega^{d/2}$ of height $d/2$, we let $\mathcal{L}_T$ denote the restriction of $\mathcal{L}$ to $L_T = T \times [0, B]$. By the above lemmas $\mathcal{L}_T$ is a finite monotonic graph which satisfies W. We now show that if T is $(d/2, n)$-universal then $\mathcal{L}_T$ is (W, n)-universal. This requires defining a morphism from G to $\mathcal{L}_T$ whenever G satisfies W, which is non-trivial and the object of the second section.

2 Universality of $\mathcal{L}_T$

Recall from Chapter 5 that we obtained a morphism from an even graph G to an adequate monotonic graph by considering relevant odd occurrences over paths in G. We extend this idea to the current setting by defining a notion of relevant occurrences which take the weights (and their boundedness) into account. We recall that B is fixed to $B = (n-1)N$.

Window decompositions. A *window decomposition* $\mathcal{I}$ (of ω) is a finite or infinite sequence $\mathcal{I} = I_0 I_1 \dots$ of disjoint intervals of the form $I_j = [a_j, b_j]$ with $b_{j+1} = a_j + 1$ if it is defined, and $\bigcup_j I_j = \omega$. Stated differently, it is a partition of ω into intervals (ordered naturally). Given a word of weights $w_{\text{wei}} = t_0 t_1 \dots \in \mathbb{Z}^\omega$, we say that it is *$B$-bounded* over $\mathcal{I}$ if for each interval I_j,

$$\forall i_0, i_1 \in I_j, \qquad \textstyle\sum_{i=i_0}^{i_1-1} t_i \leqslant B.$$

Given a window decomposition $\mathcal{I}$ and an infinite word of priorities $w_{\text{prio}} = p_0 p_1 \dots$, we define its $\mathcal{I}$-occurrences as the finite or infinite word (according to finiteness of $\mathcal{I}$) by

$$w_{\text{prio},\mathcal{I}} = (\max_{i \in I_0} p_i)(\max_{i \in I_1} p_i) \dots$$

Recall from Chapter 5 that in a finite even graph relevant odd occurrences are bounded by $n-1$, via a simple pumping argument (Lemma 5.2). The following is an analogous result in the more difficult setting of mean-payoff parity games.

Lemma 7.3 (Bounded windows in finite graphs satisfying W)

Let G be a graph of size $\leqslant n$ over V which satisfies W and let $\pi : v_0 \overset{w}{\rightsquigarrow}$ be an infinite path in G. There exists a window decomposition $\mathcal{I}$ such that w_{wei} is B-bounded over $\mathcal{I}$ and moreover $w_{\text{prio},\mathcal{I}}$ has at most $2n$ relevant occurrences of each odd priority.

The bound $2n$ could probably be reduced to n by a more careful examination but this will have no influence: it only matters that there is such a bound.

Proof. We denote $\pi : v_0 \xrightarrow{(p_0,t_0)} v_1 \xrightarrow{(p_1,t_1)} \dots$ and therefore $w = (p_0,t_0)(p_1,t_1)\dots$. We build $\mathcal{I}$ greedily by taking intervals as large as possible such that $w_{\text{wei}} = t_0 t_1 \dots$ is B-bounded over $\mathcal{I}$. Stated differently, we pick $\mathcal{I} = I_0 I_1 \dots$ to be the unique window over which w_{wei} is B-bounded and satisfying for each interval $I_j = [a_j, b_j]$ where b_j is finite that there exists $i_{j,0} \in I_j$ such that

$$\textstyle\sum_{i=i_{j,0}}^{b} t_i > B,$$

or in words: $b_j + 1$ could not be added to I_j without breaking B-boundedness. Note that $\mathcal{I}$ could be finite or infinite.

By a straightforward pumping argument, since $\sum_{i=i_{j,0}}^{b_j} t_i > B$ and since $B = (n-1)N$ and weights are upper-bounded by N, there are for each j such that I_j is finite two indices $i_{j,1}$ and $i_{j,2}$ belonging to $[i_{j,0}, b_j] \subseteq I_j$ such that $\pi_{[i_{j,1}, i_{j,2}]}$ is a cycle of positive sum.

We assume towards contradiction that $w_{\text{prio},\mathcal{I}}$ has $2n+1$ relevant occurrences of some odd priority p: there exist $j_0 < j_1 < \dots < j_{2n}$ such that for all $k \in [0, 2n]$,

$$\max_{i \in I_{j_k}} p_i = p_{i_k} = p$$

for some indices $p_{i_k} \in I_{j_k}$, and moreover for all $i \leqslant i_{2n}$ it holds that $p_i \leqslant p$. There must be two indices $k_0, k_1 \in [0, 2n]$ with $k_0 + 1 < k_1$ such that $v_{j_{k_0}} = v_{j_{k_1}}$. For simplicity, we let u and u' denote respectively $v_{j_{k_0}}$ and $v_{i_{j_{k_0+1},1}}$ defined above. We have

$$\pi_{[j_{k_0}, j_{k_1}]} : u \overset{w_1}{\rightsquigarrow} u' \overset{w_2}{\rightsquigarrow} u' \overset{w_3}{\rightsquigarrow} u,$$

where $w_{2,\text{wei}}$ has positive sum. Therefore for some large enough s the cycle obtained by repeating s times the cycle around u', formally

$$\pi'_s : u \overset{w_1}{\rightsquigarrow} \underbrace{u' \overset{w_2}{\rightsquigarrow} u' \overset{w_2}{\rightsquigarrow} \dots \overset{w_2}{\rightsquigarrow} u'}_{s \text{ times}} \overset{w_3}{\rightsquigarrow} u$$

has positive sum. Moreover its first priority is p which is odd and all priorities are $\leqslant p$ and therefore $(\pi'_s)^\omega$ does not satisfy W. □

Truncated vector of odd occurrences. Note that in a (finite or infinite) word of priorities $w = p_0p_1\dots$ there are only relevant occurrences of priorities $\geqslant p_0$. Therefore in the current setting it is natural to consider the *truncated vector of odd occurrences* which is defined over $w = p_0p_1\dots$ by restricting to occurrences of odd priorities greater than p_0, formally

$$\mathrm{occ}^{\mathrm{tr}}(w) = (\mathrm{occ}_{d-1}(w), \mathrm{occ}_{d-3}(w), \dots, \mathrm{occ}_{\lceil p_0\rceil^{\mathrm{odd}}}(w)).$$

Note in particular that its last index is given by $p_{\mathrm{occ}^{tr}(w)} = \lceil p_0\rceil^{\mathrm{odd}}$.

Definition of ϕ. Given a word $w = (p_0,t_0)(p_1,t_1)\dots \in C^\omega$ and a window decomposition $\mathcal{I} = I_0I_1\dots$ we let $\phi(w,\mathcal{I}) \in (\omega+1)^{\leqslant d/2}\times(\omega+1)$ be given by

$$\phi(w,\mathcal{I}) = \Big(\mathrm{occ}^{\mathrm{tr}}(w_{\mathrm{prio},\mathcal{I}}), \sup_{i_1\in I_0}\textstyle\sum_{i=0}^{i_1-1} t_i\Big).$$

In words $\phi(w,\mathcal{I})$ is comprised of two components: the first gives (truncated) relevant odd occurrences among those which are maximal for the I_j's, and the second gives the peak of the profile within the first window. Note that both of these may be infinite (or have infinite coordinates) in general.

On the example word below (which belongs to W),

$$w = \underbrace{(3,4)(1,-4)(6,5)(2,-1)(3,8)}_{I_0}\,\underbrace{(2,3)(5,-12)}_{I_1}\,\underbrace{(9,-1)(8,2)(9,-4)(7,18)(9,0)(9,0)(9,0)\dots}_{I_3}$$

where $\mathcal{I} = [0,4][5,6][7,\omega]$ and if $d = 10$, we have

$$\phi(w,\mathcal{I}) = \big(\mathrm{occ}^{\mathrm{tr}}(659), \sup\{4,0,5,4\}\big) = \big((1,0),5\big).$$

Indeed in this case, $\max_{i\in I_0} p_i = 6$ therefore only relevant occurrences of 9 and 7 are accounted for; there is only one occurrence of 9 as the maximum priority over I_3 and no occurrence of the priority 7; and lastly $5 = 4-4+5$ is the peak of the profile over I_0 (note that $4-4+5-1+8$ is not a profile of I_0 by definition, since the sum ranges over integers up to i_1-1).

Note that if w_{wei} is B-bounded over $\mathcal{I}$ then in particular the second component of $\phi(w,\mathcal{I})$ is $\leqslant B$.

Lemma 7.4 (Main technical ingredient for universality)

Let $w = (p_0,t_0)w'$ and let $\mathcal{I}$ be a window decomposition over which w_{wei} is B-bounded and such that $\mathrm{occ}^{tr}(w_{\mathrm{prio},\mathcal{I}})$ is finite. There exists a window decomposition $\mathcal{I}'$ such that w'_{wei} is B-bounded over $\mathcal{I}'$ and the edge e given by

$$e = \phi(w,\mathcal{I}) \xrightarrow{(p_0,t_0)} \phi(w',\mathcal{I}')$$

belongs to $\mathcal{L}$.

In the proof below $\mathcal{I}'$ is always naturally defined from $\mathcal{I}$ simply by restricting to coordinates $\geqslant 1$. The formal definition differs according to whether $I_0 = \{0\}$ or not which is why the proof is slightly tedious.

Proof. We let $w = (p_0,t_0)(p_1,t_1)\dots$ and we use $p_{\mathcal{I},j}$ to denote $\max_{i\in I_j} p_i$. We also let $w' = (p_1,t_1)(p_2,t_2)\dots = (p'_0,t'_0)(p'_1,t'_1)\dots$ and use the same notation $p'_{\mathcal{I}',j}$ once $\mathcal{I}' = I'_0I'_1\dots$ is fixed. We let $(u,r) = \phi(w,\mathcal{I})$ and $(u',r') = \phi(w',\mathcal{I}')$ (again, once $\mathcal{I}'$ is fixed this makes sense).

We distinguish three cases according to $\mathcal{I}$.

- If there is $i \geqslant 1$ such that $i \in I_0$ and $p_i = p_{\mathcal{I},0}$. Then we let $\mathcal{I}'$ be given by

  $$I'_0 = (I_0 - 1) \cap \omega \neq \varnothing \qquad \text{and} \qquad I'_j = I_j - 1$$

  where it makes sense (meaning I_j is defined). It is clear that w'_{wei} is B-bounded over $\mathcal{I}'$ since w_{wei} is B-bounded over $\mathcal{I}$.

  Then we have $w_{\text{prio},\mathcal{I}} = w'_{\text{prio},\mathcal{I}'}$ therefore $u = u'$. Now let $i'_1 \in I'_0$ be such that $r' = \sum_{i'=0}^{i'_1-1} t'_{i'} = \sum_{i=1}^{i'_1} t_i$. Then $i'_1 + 1 \in I_0$ therefore

  $$r = \max_{i_1 \in I_0} \textstyle\sum_{i=0}^{i_1-1} t_i \geqslant t_0 + r'$$

  hence e is an edge of type $(2b)$.

- If $|I_0| \geqslant 2$ and for all $i \in I_0$ with $i \geqslant 1$ we have $p_i < p_0$. In this case we let $\mathcal{I}'$ be defined just like above, and we have $w_{\text{prio},\mathcal{I}} = p_0 w''$ and $w'_{\text{prio},\mathcal{I}'} = p'w''$ for some finite or infinite word of priorities w'' and with $p_0 > p'$. Note that $u = \text{occ}^{\text{tr}}(p_0 w'')$ is such that $p_u = \lceil p_0 \rceil^{\text{odd}}$.

  All relevant occurrences of priorities $\geqslant p_0$ in $p'w''$ are also relevant in $p_0 w''$. Therefore if p_0 is even we have $u \geqslant_{p_0+1} u'$ and e is of type $(1a)$. If p_0 is odd, 0 is an additional occurrence of p_0 in $p_0 w''$ therefore $u >_p u'$ and e is of type $(1b)$.

- If $I_0 = \{0\}$. Then we let $\mathcal{I}'$ be given by $I'_j = I_{j+1}$ that is $\mathcal{I}' = I_1 I_2 \ldots$ and it is clear that w'_{wei} is B-bounded over $\mathcal{I}'$. In this case we have $w_{\text{prio},\mathcal{I}} = p_0 w'_{\text{prio},\mathcal{I}'}$ thus all relevant occurrences of priorities $\geqslant p_0$ in $w'_{\text{prio},\mathcal{I}'}$ are also relevant in $w_{\text{prio},\mathcal{I}}$, and again $p_u = \lceil p_0 \rceil^{\text{odd}}$: we conclude just like in the previous case. □

We now let

$$\phi(w) = \min\{\phi(w, \mathcal{I}) \mid w_{\text{wei}} \text{ is } B\text{-bounded over } \mathcal{I}\}$$

where the minimum is taken lexicographically. By Lemma 7.3 it holds that $\phi(w) \in [0, 2n]^{\leqslant d/2} \times [0, B]$ for colourations w of graphs of size $\leqslant n$ which satisfy W.

Given a vertex v in such a graph we naturally let

$$\phi_G(v) = \max_{v \overset{w}{\rightsquigarrow} \text{ in } G} \phi(w).$$

We define $T(G) \subseteq [0, 2n]^{\leqslant d/2}$ as the projection of $\phi_G(V)$ on its first coordinate; note that $\phi_G(v) \in L_{T(G)}$ by definition.

Universality. We now have all the tools in hands to prove our main result in this chapter.

Theorem 7.1 (Universality of the construction)

If G is a finite graph of size $\leqslant n$ satisfying W then ϕ_G defines a graph morphism from G to $\mathcal{L}_{T(G)}$. In particular if T is an n-universal tree then $\mathcal{L}_T$ is a (W, n)-universal monotonic graph.

Proof. For the second statement it suffices to compose on the right with the natural morphism of $\mathcal{L}_{T(G)}$ into $\mathcal{L}_T$ therefore we concentrate on the first statement: let G be such a finite graph and let $e = v \xrightarrow{(p,t)} v'$ in G.

Let $v' \overset{w'}{\rightsquigarrow}$ in G be such that $\phi_G(v') = \phi(w')$ and let $\mathcal{I}$ be a window decomposition over which w_{wei} is B-bounded and which is minimal in the sense that $\phi(w) = \phi(w, \mathcal{I})$. Then Lemma 7.4 gives a window decomposition $\mathcal{I}'$ such that w' is B-bounded over $\mathcal{I}'$ and $\phi(w,\mathcal{I}) \xrightarrow{(p,t)} \phi(w',\mathcal{I}')$ is an edge in $\mathcal{L}$.

Now we have in $\mathcal{L}$

$$\phi(v) \geqslant \phi(w,\mathcal{I}) \xrightarrow{(p,t)} \phi(w',\mathcal{I}') \geqslant \phi(w') = \phi(v'),$$

and therefore by left and right composition in $\mathcal{L}$, $\phi(v) \xrightarrow{(p,t)} \phi(v')$ belongs to $\mathcal{L}$ and thus also to its restriction $\mathcal{L}_{T(G)}$ to $T(G)$. □

8 Finite monotonic graphs for multi mean-payoff games

Context and contribution. In this short chapter, we study the lim sup variant of multi mean-payoff games which was introduced by Velner, Chatterjee, Doyen, Henzinger, Rabinovich and Raskin [VCD+15]. We refer to the introduction for a survey of multi mean-payoff games.

The lim sup variant (formally defined below) is well-known to be tractable, and a $O(mn^2dN)$ algorithm was presented in [VCD+15], essentially by reduction to nd calls to a procedure for $\mathrm{MP}^{\leqslant 0}$. The lim inf variant however is NP-complete[1] [VCD+15], and known techniques [CV12; CJL+17] for solving it are more involved.

We give a simple construction of a universal monotonic graph, improving the runtime bound to $O(mn\log(n)dN)$, thus removing a multiplicative $n/\log(n)$ factor. The first section gives the main technical result, which roughly states that two quantifiers can be exchanged. The second section shows how combine (universal) monotonic graphs to build on such a result. The approach we took in [AFGL+21] is exactly the same, but in the vocabulary of separating automata.

Notations. We fix non-negative integers N and d. The set of colours is $C = [-N, N]^d$. We write letters as $t = (t^0, t^1, \dots, t^{d-1}) = (t_j)_{j \in [0,d-1]} \in C$ and words as $w = t_0 t_1 \dots$. Given a word $w = t_0 t_1 \dots$ and $j \in [0, d-1]$ we write $w^j = t_0^j t_1^j \dots$ for the projection of w on the j-th coordinate.

We let $\mathrm{MP}^{j,\leqslant 0}$ denote the threshold mean-payoff objective over the j-th coordinate, formally

$$\begin{array}{rcl} \mathrm{MP}^{j,\leqslant 0} & = & \{t_0 t_1 \dots \in C^\omega \mid \limsup_n \frac{1}{n} \sum_{i=0}^{n-1} t_i^j \leqslant 0\} \\ & = & \{t_0 t_1 \dots \in C^\omega \mid w^j \in \mathrm{MP}^{\leqslant 0}\}. \end{array}$$

The multi mean-payoff objective is the union of the $\mathrm{MP}^{j,\leqslant 0}$'s, formally

$$\begin{array}{rcl} W & = & \bigcup_{j=0}^{d-1} \mathrm{MP}^{j,\leqslant 0} \\ & = & \{w \in C^\omega \mid \exists j, \limsup_n \frac{1}{n} \sum_{i=0}^{n-1} t_i^j \leqslant 0\}. \end{array}$$

It is positionally determined for Eve by concavity of $\mathrm{MP}^{\leqslant 0}$ in the lim sup semantic, as proved by Kopczyński [Kop06]. It turns out that satisfying W for a finite C-graph is quite easy to understand.

Given a C-graph G, we let G^j denote the $[-N, N]$-graph obtained by projecting on the j-th coordinate, formally

$$v \xrightarrow{t} v' \text{ in } G^j \qquad \iff \qquad \exists t' \in C, t'^j = t \text{ and } v \xrightarrow{t'} v' \text{ in } G.$$

The *mean* of a finite word $w \in \mathbb{Z}$ is its sum divided by its length, and the terminology is extended to paths.

[1] In [VCD+15], coNP-completeness is established, but the point of view taken is that of the opponent.

1 Strongly connected graphs satisfying W

A graph is *strongly connected* if for any $v, v' \in V$ there is a path from v to v'. The following states a collapse result for strongly connected graphs: if W is satisfied then one of the $\mathrm{MP}^{j,\leqslant 0}$ is satisfied.

Theorem 8.1 (Collapse for strongly connected graphs)

Let G be a finite strongly connected graph satisfying W. There exists $j \in [0, d-1]$ such that G satisfies $\mathrm{MP}^{j,\leqslant 0}$.

A similar result was given in [VCD+15] (Lemma 8 therein), our proof is a bit simpler.

Proof. Let $n = |V|$ and assume for contradiction that for each $j \in [0, d-1]$, G does not satisfy $\mathrm{MP}^{j,\leqslant 0}$, or equivalently G^j is not bounded: it has a positive cycle which gives a cycle $\pi_j : v_j \overset{w_j}{\rightsquigarrow} v_j$ in G such that w_j^j has positive sum.

Let $j \in [0, d-1]$ and let $\pi : v_0 \overset{w}{\rightsquigarrow} v$ in G be any finite path. Thanks to strong connectedness there is a path from v to v_j, and we consider

$$\pi'_k : v_0 \overset{w}{\rightsquigarrow} v \overset{w'}{\rightsquigarrow} \underbrace{v_j \overset{w_j}{\rightsquigarrow} v_j \overset{w_j}{\rightsquigarrow} \dots \overset{w_j}{\rightsquigarrow} v_j}_{k \text{ times}}$$

for $r \in \omega$, whose coloration is $u_k = ww'(w_j)^k$. The sum of u_k^j is $s + s' + ks_j$ and its length is $r + r' + kr_j$ with obvious notations, and therefore its mean goes to $s_j / r_j \geqslant 1/n$ when k grows large. Therefore k can be picked large enough so that the mean of π'_k on the j-th coordinate is $\geqslant 1/(2n)$.

Starting from the empty path and iterating this process cyclically for $j = 0, 1, \dots, d-1, 0, 1, \dots$ constructs an infinite path whose mean-payoff is $\geqslant 1/(2n)$ on every coordinate, which contradicts the assumption. □

We now exploit this property to construct succint (W, n)-universal monotonic graphs.

2 A W-universal monotonic graph

Direct sum of graphs. Given a finite sequence of graphs $G_0, \dots, G_{r-1}$ over $V_0, \dots, V_r$ we define their *direct sum* G to be the C-graph over $\{0\} \times V_0 \cup \dots \cup \{r-1\} \times V_{r-1}$ given by

$$(i, v) \xrightarrow{c} (i', v') \text{ in } G \qquad \iff \qquad i > i' \text{ or } (i = i' \text{ and } v \xrightarrow{c} v' \text{ in } G_i).$$

This is illustrated in Figure 8.1. We denote G by $G_0 \oplus \dots \oplus G_{r-1}$ or $\bigoplus_{i=0}^{r-1} G_i$.

Informally, G is obtained by putting copies of the G_i's next to one another and adding all edges from right to left.

Note that this operation is associative up isomorphism.

Lemma 8.1 (Closure properties of $\oplus$)

(i) If $G_0, \dots, G_{r-1}$ satisfy a prefix-decreasing objective W then so does their direct sum.

(ii) If $\mathcal{L}_0, \dots, \mathcal{L}_{r-1}$ are monotonic then so is their direct sum.

(iii) If G embeds in G_{i_0} for some i_0 then it embeds in $\bigoplus_i G_i$.

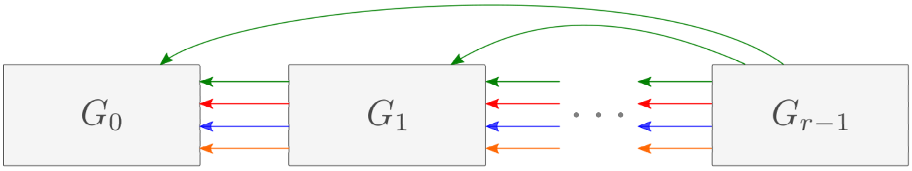


Figure 8.1: Illustrating the direct sum. Many edges are not depicted for clarity.

Proof. (i) A path in $\bigoplus_i G_i$ eventually remains in some G_i therefore it satisfies W by prefix-decreasingness.

(ii) The order over $\mathcal{L} = \bigoplus_i \mathcal{L}_i$ is naturally given by

$$(i,\ell) \geqslant (i',\ell') \qquad \Longleftrightarrow \qquad i > i' \text{ or } (i = i' \text{ and } \ell \geqslant \ell' \text{ in } \mathcal{L}_i).$$

We show-left composition: assume $(i,\ell) \geqslant (i',\ell') \xrightarrow{c} (i'',\ell'')$ in $\mathcal{L}$. If $i > i''$ the edge $(i,\ell) \xrightarrow{c} (i'',\ell'')$ belongs to $\mathcal{L}$ by definition and otherwise $i = i''$ therefore $i = i' = i''$ and left composition in $\mathcal{L}_i$ concludes. The proof of right composition follows the same lines.

(iii) Let $\phi_{i_0} : V \to V_{i_0}$ be a morphism from G to G_{i_0}. We extend it to a morphism ϕ into the direct sum by the formula $\phi(v) = (i_0, \phi_{i_0}(v))$. If $v \xrightarrow{c} v'$ in G then $\phi_{i_0}(v) \xrightarrow{c} \phi_{i_0}(v')$ in G_{i_0} therefore $\phi(v) \xrightarrow{c} \phi(v')$ in the direct sum.

□

Therefore if $\mathcal{L}_0, \dots, \mathcal{L}_{r-1}$ are finite monotonic graphs satisfying W then so is their direct sum. Note that this can be directly extended to ordinal-indexed direct sums of (potentially infinite) well-monotonic graphs, but doing so is not required here.

Lemma 8.2 (Construction for strongly connected graphs)

For each n there is a monotonic graph $\mathcal{L}_n^{\text{sc}}$ of size $d(n-1)N$ which satisfies W and embeds all strongly connected graphs of size $\leqslant n$ satisfying W.

Proof. For each $j \in [0, d-1]$ we let $\mathcal{L}_n^j$ be the monotonic graph over $[0,(n-1)N]$ given by

$$\ell \xrightarrow{t} \ell' \qquad \Longleftrightarrow \qquad t^j \leqslant \ell - \ell'.$$

We know by Chapter 6 that $\mathcal{L}_j$ embeds any C-graph of size $\leqslant n$ satisfying $\mathrm{MP}^{j,\leqslant 0}$, and that it satisfies $\mathrm{MP}^{j,\leqslant 0}$ therefore it satisfies W.

We conclude thanks to Theorem 8.1 and Lemma 8.1 that the direct sum $\mathcal{L}_n^{\text{sc}} = \bigoplus_{j=0}^{d-1} \mathcal{L}_n^j$ embeds all strongly connected C-graphs of size $\leqslant n$ which satisfy W. □

From there we may already obtain a (W,n)-universal monotonic graph of size $dn(n-1)N$ by considering the direct product of n copies of $\mathcal{L}_n^{\text{sc}}$. We will actually give a slightly more succint construction which replaces the quadratic dependency in n with $n \log n$ by applying the technology developped for parity games.

Universal words. We consider finite words of non-negative integers $u \in \omega^*$. The *sum* of such a word is the sum of its letters. A *morphism* from u to u' is[2] an increasing $f : |u| \to |u'|$ such that for all $i \in |u|$ we have $u_i \leqslant u'_{f(i)}$.

[2] Recall that $|u|$ denotes the length of the word u, not its sum.

Observe a strong parallel with Chapter 5: words and their morphisms are exactly equivalent to trees of height 2 and their morphisms[3] by the bijection

$$u_0 u_1 \dots u_{r-1} \qquad \leftrightarrow \qquad (\{0\} \times |u_0|) \cup (\{1\} \times |u_1|) \cup \dots \cup (\{r-1\} \times |u_{r-1}|).$$

(Any tree of height 2 can be written in this form up to isomorphism.) The correspondence is illustrated in Figure 8.2.

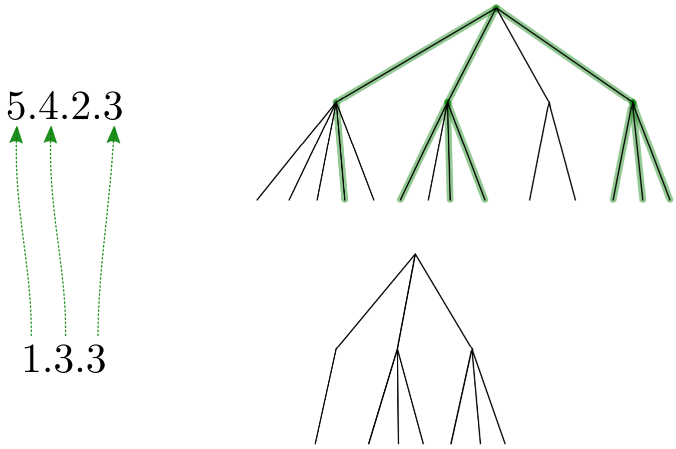


Figure 8.2: On the left, two sequences and a morphism between them. On the right, corresponding trees of height 2 and a tree morphism.

A word $w \in \omega^*$ is n-universal if it embeds all words with sum $\leqslant n$. Instanciating the quasipolynomial construction of a universal tree from [JL17] (see Chapter 5) to height 2 gives a succint construction of a n_k-universal word u_k for $n_k = 2^{k+1} - 1$ which can also be described inductively by

$$u_0 = 1 \qquad \text{and} \qquad u_k = u_{k-1} n_k u_{k-1}.$$

The first few values are given by

$$u_0 = 1 \qquad u_1 = 1.3.1 \qquad u_2 = 1.3.1.5.1.3.1 \qquad u_3 = 1.3.1.5.1.3.1.9.1.3.1.5.1.3.1$$

One may prove directly by induction over k that u_k is n_k-universal. First, this is clear for u_0. Now if u has sum $\leqslant n_k = 2^{k+1} - 1$ then u can be written as $u = u' u_i u''$ for some i where u' and u'' have sum $\leqslant n_k$. Therefore both u' and u'' embed via f' and f'' in u_{k-1} and these embeddings are easily merged into a morphism from u to u_k.

It is also easy to see by induction that the sum of n_k is upper bounded by $k2^{k+1} \leqslant n_k \log n_k$. We write $u_k = u_{k,0} u_{k,1} \dots$. We are now ready to state the main result.

Theorem 8.2 (Succint universal monotonic graph for multi-energy games)

For each $k \in \omega$ the monotonic graph

$$\mathcal{L} = \bigoplus_{i \in |u_k|} \mathcal{L}^{\text{sc}}_{u_{k,i}}$$

is (W, n_k)-universal. It has size $\leqslant d n_k \log(n_k) N$.

[3] Existence of a morphism between sequences is equivalent to existence of a morphism between the corresponding trees (number of morphisms however differ).

This implies for all n a (W, n)-universal monotonic graph of size $\leqslant 2dn\log(n)N$ simply by rounding n up to the next integer of the form $2^{k+1} - 1$.

Proof. The facts that $\mathcal{L}$ is monotonic and that it satisfies W follow from Lemmas 8.1 and 8.2. Let G be a graph over V satisfying W of size $\leqslant n_k$ and consider a topological ordering $g : V \to [0, r-1]$ of its strongly connected components, formally

$$\big[v \to v' \text{ in } G \implies g(v) \geqslant g(v')\big] \qquad \text{and} \qquad \big[g(v) = g(v') \implies v \rightsquigarrow v' \rightsquigarrow v \text{ in } G\big].$$

For each $i \in [0, r-1]$, the restriction G_i of G to $V_i = f^{-1}(i)$ is a strongly connected graph satisfying W and therefore it embeds in $\mathcal{L}^{\mathrm{sc}}_p$ provided $p \geqslant |V_i|$ by Lemma 8.2.

Now by universality of u_k there is a word-morphism f from $|V_0||V_1|\ldots|V_{r-1}|$ to u_k, which is a map $f : [0, r-1] \to |u_k|$ satisfying for all $i \in [0, r-1]$ that $|V_i| \leqslant u_{k,f(i)}$. Thus there is for each i a graph-morphism ϕ_i from G_i to $\mathcal{L}^{\mathrm{sc}}_{u_{k,f(i)}}$.

Hence the morphism $\phi : V \to \mathcal{L}$ defined over V_i by $\phi(v) = (f(i), \phi_i(v))$ defines a graph morphism: the image of an edge $V_i \xrightarrow{V}_{i'}$ belongs to the $f(i)$-th component of the sum if $i = i'$ and it otherwise $i > i'$ by definition of g therefore the edge belongs to $\mathcal{L}$. □

We conclude that multi mean-payoff games can be solved by value iteration in $\mathcal{L}$ with complexity

$$\text{runtime: } O(mdn\log(n)N) \qquad\qquad \text{space required: } O\big(n\log(dnmN)\big).$$

This saves a factor $n/\log n$ in the runtime over the algorithm of [VCD+15].

Conclusion and perspectives for Part II

We have studied value iteration algorithms, which solve games by Kleene iteration when given a monotonic universal graph. We have seen that such graphs may be obtained without blow-up from strongly separating automata, and studied two different variants of the value iteration algorithm.

We showed that graphs satisfying parity (or equivalently, winning positional strategies) naturally induce ordered trees obtained by counting maximal numbers of relevant odd priorities along their paths. This allowed to connect to universal trees, for which we presented the known upper and lower bounds. There remains a linear gap between the two bounds, we would expect that the lower bound might be improved because of the somewhat brutal minoration indicated by $(*)$ on page 137; however we have not been able to close this gap so far.

For mean-payoff games, we saw that the BCDGR algorithm, which has state-of-the-art quasipolynomial $O(mnN)$ runtime, is easily described in our framework. We gave a $N^{1-1/n}$ lower bound, and a $O(nN)$ construction which matches this dependency in N when n is fixed. Again, this leaves a linear n multiplicative gap, which we have not been able to close. We also studied the case where the set of weights is fixed with cardinality k, and gave upper and lower bounds of roughly n^k.

For mean-payoff parity games, we showed how to formulate the state-of-the-art value iteration algorithm of [DJL18] in our framework. We also explained differences between the two approaches: ours is completely asymmetric and therefore more direct. Finally, for multi mean-payoff games in the $\limsup$ semantic, we showed how to combine universal monotonic graphs, and imported an idea from parity games (universal sequences) to improve on the bound of [VCD+15].

Which structures are modular? We would like to further study what structural properties of the natural construction $\mathcal{L}_{[0,(n-1)N]}$ for mean-payoff games enables the combinations with parity games[4] (Chapters 7) and/or with other such structures (Chapter 8). Can these constructions be generalised to other meaningful subclasses of monotonic graphs?

Note that we know from the work of Chatterjee, Henzinger and Piterman [CHP07] that solving disjunctions of two parity conditions is NP-complete, therefore one cannot expect to come up with such direct constructions for combining quasipolynomial constructions for parity games.

[4] One can easily imagine using a similar construction for a disjunctions with an arbitrary lexicographical product. Can we start with the union of $\mathrm{MP}^{\leqslant 0}$ and a lexicographical product?

Part III

Beyond value iterations

Strategy improvement with fixpoint valuations 9

The three chapters of Part III are more independent and exploratory, and we make no general discussion: each has its own introduction and/or conclusion.

This chapter is concerned with strategy improvement algorithms; we refer to the general introduction for additional context. Our main result is the following: a valuation which is computed by a monotonic graph is fit for strategy improvement if and only if it is positional for Adam. The first section introduces the necessary definitions, proves the theorem, and compares with existing related work. The second section discusses concrete applications and perspectives.

1 A generic framework for strategy improvement

In this section, we first define strategy improvement algorithms, then introduce fixpoint valuations and prove our main theorem, and finally compare with other generic frameworks from the literature.

1.1 Strategy improvement

In our framework, it is crucial that Adam is the *improver*: the algorithm iterates over Adam positional strategies $\tau_0, \tau_1, \ldots$, which induce graphs $G_{\tau_0}, G_{\tau_1}, \ldots$ controlled by his opponent Eve.

Graphs controlled by Eve. In contrast with the rest of the thesis, we therefore now see (finite) graphs as arenas which are controlled by Eve in this chapter.

Consequently given a valuation val and a graph G we now let

$$\mathrm{val}_G(v) = \inf_{v \xrightarrow{w} \text{ in } G} \mathrm{val}(w),$$

and given a completely monotonic graph $\mathcal{L}$, a finite graph G over V and a progress measure ϕ we now have

$$\mathrm{Upd}^{\mathcal{L}}_G(\phi)(v) = \min_{v \xrightarrow{c} v' \text{ in } G} \rho(\phi(v'), c).$$

Definitions over general arenas remain unchanged.

Switches. Fix a finite arena G over V. We describe strategy improvement algorithms from the point of view of Adam; the algorithm iteratively improves on a uniform positional strategy τ :

$V_{\text{Adam}} \to E$ for Adam. Given such a strategy τ_0, and given an edge $e_1 = v \xrightarrow{c_1} v'_1$ in G, we let τ_1 denote the uniform positional Adam strategy given by

$$\tau_1(v) = e_1 \qquad \text{and} \qquad \forall v' \neq v, \tau_1(v') = \tau_0(v').$$

We say that τ_1 is obtained from τ_0 by *switching with* e_1.

We now let val : $C^\omega \to X$ be an arbitrary valuation. Given a uniform positional strategy τ for Adam, we say that e' *defines an improving switch* for τ if the strategy τ' obtained by switching with e' satisfies

$$\text{val}_{G_\tau} < \text{val}_{G_{\tau'}}.$$

Axioms. We say that val is *fit for strategy improvement* if the three following conditions are satisfied.

(A) Graph tractability: given a graph G (controlled by Eve) one may compute val_G efficiently.

(B) Co-positionality: the valuation is uniformly positionally determined for Adam.

(C) Existence of improving switches: for any uniform positional strategy τ which is not val-optimal, there exists an edge e' which defines an improving switch.

Graph tractability corresponds to efficient computation of an optimal counter strategy σ, and co-positionality with the existence of an optimal uniform positional strategy τ for Adam. The existence of improving switches is key for applying the precise mechanism of strategy improvement algorithms.

Strategy improvement. If the three conditions hold, one may run a strategy improvement algorithm over a finite arena G, which is described as follows.

1. Initialise τ as an arbitrary uniform positional strategy for Adam.

2. For each edge e', check if it defines an improving switch for τ.

3. If there is no such edge, then τ is necessarily optimal by (C) and the iteration stops.

4. Otherwise, choose an improving switch e' for τ, replace τ with the strategy τ' obtained by switching with e', and iterate from step 2.

Thanks to graph tractability, each iteration is effective. The choice of the improving switch in step 4 is often called a *switching policy*. For known valuations which are fit for strategy improvement, it is generally the case that improving switches are *combinable*, in the sense that applying several at the same time always improves the valuation.

Moreover, step 2 can usually be performed with only a single call to the procedure for solving graphs: one may determine improving switches directly from val_{G_τ}, without having to explicitly compute $\text{val}_{G_{\tau'}}$. This allows to save a factor m in the complexity of performing an iteration.

If the set of values X is finite, then $n|X|$ provides an upper bound on the number of iterations, which is usually very large; strategy improvement algorithms generally display a much faster convergence.

1.2 Fixpoint valuations

We say that val : $C^\omega \to L$ is a *fixpoint valuation* if L is the set of vertices of a completely monotonic graph $\mathcal{L}$ such that for any finite arena G,

$$\mathrm{val}_G = \psi_G^{\mathcal{L}}.$$

In words, val coincides with the $\mathcal{L}$-evaluation $\psi_G^{\mathcal{L}}$ (which is defined to be the least fixpoint of $\mathrm{Upd}_G^{\mathcal{L}}$, see Chapter 1) over finite arenas. In particular, a fixpoint valuation is always positional for Eve.

Theorem 4.1 states that if $\mathcal{L}$ is finite then $\mathrm{val}^{\mathcal{L}}$ is a fixpoint valuation (in this case, the above identity even holds over arbitrary arenas). We have seen many other examples of fixpoint valuations; we refer to the next section for a discussion. Our main technical result in this chapter is the following.

Theorem 9.1 (Strategy improvement with fixpoint valuations)

A fixpoint valuation which is uniformly positionally determined for Adam over finite arenas has improving switches.

Stated differently any fixpoint valuation which is tractable over graphs and co-positionally determined is fit for strategy improvement.

Proof. Let val : $C^\omega \to L$ be a fixpoint valuation associated to the completely monotonic graph $\mathcal{L}$ over L, and consider a finite arena G over V. Let $\tau_0 : V_{\mathrm{Adam}} \to E$ be a uniform positional strategy for Adam which is not optimal, which rewrites as

$$\psi_{G_{\tau_0}}^{\mathcal{L}} < \psi_G^{\mathcal{L}}.$$

For convenience we let ψ_0 and ψ respectively denote the left hand-side and right hand-side and $G_0 = G_{\tau_0}$ (which is controlled by Eve). By Knaster-Tarski ψ_0 is not a prefixpoint of Upd_G (it is smaller than the least fixpoint) : there is $v \in V$ such that

$$\mathrm{Upd}_G(\psi_0)(v) > \psi_0(v) = \mathrm{Upd}_{G_0}(\psi_0)(v).$$

Now if $v \in V_{\mathrm{Eve}}$ then $\mathrm{Upd}_G(\phi)(v)$ and $\mathrm{Upd}_{G_0}(\phi)(v)$ coincide for any progress measure ϕ (and in particular for ψ_0), therefore it must be that $v \in V_{\mathrm{Adam}}$ and the above rewrites as

$$\max_{v \xrightarrow{c_1} v'_1} \rho(\psi_0(v'_1), c_1) > \psi_0(v) = \rho(\psi_0(v'_0), c_0),$$

where $e_0 = v \xrightarrow{c_0} v'_0 = \tau(v)$. We let $e_1 = v \xrightarrow{c_1} v'_1$ be such that

$$\rho(\psi_0(v'_1), c_1) > \rho(\psi_0(v'_0), c_0),$$

and prove that it defines an improving switch for τ_0. We let τ_1 be obtained from τ by switching with e_1 and let $\psi_1 = \psi_{G_{\tau_1}}$. We consider the *switching arena* (see illustration in Figure 9.1) G' over V whose partition is given by $V'_{\mathrm{Adam}} = \{v\}$ and $V'_{\mathrm{Eve}} = V \setminus \{v\}$, with outgoing edges e_0 and e_1 from v and which is everywhere else identical to G_0 (and therefore also to $G_1 = G_{\tau_1}$).

There are only two uniform positional strategies τ'_0 and τ'_1 for Adam in G', respectively given by $\tau'_0(v) = e_0$ and $\tau'_1(v) = e_1$, and their graphs $G'_{\tau'_0}$ and $G'_{\tau'_1}$ respectively correspond to G_0 and G_1. Therefore the values of τ'_0 and τ'_1 are equal to those of τ_0 and τ_1 respectively, namely ψ_0 and ψ_1.

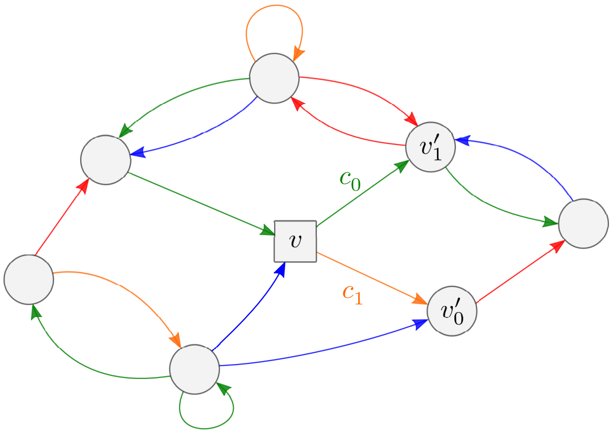


Figure 9.1: An example of switching arena G' in the proof of Theorem 9.1. Note that many vertices have a single outgoing edge: these correspond to Adam vertices in G.

Now

$$\mathrm{Upd}_G(\psi_0)(v) = \max(\rho(\psi_0(v_0'), c_0), \rho(\psi_0(v_1'), c_1)) = \rho(\psi_0(v_1'), c_1) > \rho(\psi_0(v_0'), c_0) = \psi_0(v),$$

and thus τ_0' is not optimal in G'. By co-positional determinacy of val τ_1' is therefore uniformly optimal in G' and thus

$$\psi_1 > \psi_0,$$

the sought result. □

Note from the proof that for fixpoint valuations, one may determine improving switches directly from G_τ, saving a factor m in the complexity of each iteration as explained above.

1.3 Comparisons with other generic frameworks

Frameworks similar to ours have been discussed on several occurrences. The axioms usually correspond to tractability, existence of a unique optimum, and of improving switches, and are proved by hand. The third is often the most tedious to prove. Thorough expositions can also be found in Fearnley's or Friedmann's PhD theses [Fea10b; Fri11b].

We elaborate on a paper of Costan, Gaubert, Goubault, Martel and Putot [CGG+05] which proposes a general formalism tailored for direct applications in static analysis. Their approach is more general, and their focus is different. For this discussion, we write LFP(h) for the least fixpoint of a given monotonous operator h. In their setting, they dispose of an operator f over an (arbitrary) complete lattice X, which is given by $f = \inf \mathcal{G}$, where each $g \in \mathcal{G}$ is a monotonous operator (and therefore, so is f), and such that LFP(g) can be computed efficiently. Here, $\mathcal{G}$ is typically exponential, and a partial algorithm is given for computing LFP$(f) = \inf_{g \in \mathcal{G}}$ LFP(g) without explicitly computing each LFP(g).

To instantiate our formalism into theirs, one would set $X = L^V$, the set of progress measures, and $\mathcal{G} = \{\mathrm{Upd}^{\mathcal{L}}_{G_\tau}, \tau \text{ Adam positional strategy}\}$. We would then have $f = \mathrm{Upd}^{\mathcal{L}}_G = \sup \mathcal{G}$, a supremum rather than an infimum (dualising the order does not help here since we wish to compute a least fixpoint and not a greatest fixpoint; the greatest fixpoint is the constant $\top$ progress measure). The alternation LFP$(\sup \mathcal{G})$ in our framework seems irreconcilable with theirs, where LFP$(\inf \mathcal{G})$ is computed.

Their working assumption of "lower selection" which allows to run their algorithm is trivially verified in our case, and moreover no general condition is given which guarantees convergence towards a fixpoint (only a postfixpoint is guaranteed in general). With additional assumptions (of geometrical nature; roughly, f is non-expansive), it is also proved that provided the algorithm reaches a fixpoint, it reaches the least fixpoint. All in all, the two frameworks are completely incomparable.

We wonder if there exist generic algorithms for computing arbitrary LFP$(\sup \mathcal{G})$, and would be curious to compare. However, we believe that exploiting the structure of $X = L^V$ is instrumental in our approach, and allows to state co-positionality as a natural necessary condition, which in our case turns out to be sufficient (Theorem 9.1), assuming the LFP(g)'s can be computed efficiently (graph tractability).

Last, we comment on Kozachinskiy's recent paper [Koz21a] which establishes that strategy improvements may be run with any bi-positional *continuous* valuation $\mathrm{val} : A^\omega \to \mathbb{R}$. Actually, it can be inferred from Theorem 22 in [Koz21a] that any such valuation is a fixpoint valuation, and moreover with an operator that admits a unique fixpoint. In this sense, our framework generalises Kozachinskiy's (although establishing Theorem 22 is still required, and difficult), by allowing for fixpoint valuations which do not admit unique fixpoints.

2 Applications and perspectives

2.1 Discounted valuations

We have seen in Chapter 2 that discounted valuations are fixpoint valuations, and moreover they are co-positional over finite arenas (actually even over arenas of finite degree). By the result of [Ye11], when the discount factor is fixed, one may compute values over graphs in strongly polynomial time $O\left(\frac{mn}{1-\lambda}\log(\frac{n}{1-\lambda})\right)$; this was improved to $O\left(\frac{m}{1-\lambda}\log(\frac{n}{1-\lambda})\right)$ by [HMZ13]. Therefore the discounted valuation is fit for strategy improvement. It was actually shown in [HMZ13] that there are at most the same number $O\left(\frac{m}{1-\lambda}\log(\frac{n}{1-\lambda})\right)$ of iterations, and thus the strategy improvement is strongly polynomial when λ is fixed.

As it was shown in [ZP96], finite mean-payoff games can be reduced to discounted games by taking λ sufficiently close to 1 (see also Chapter 2). Puri [Pur95] was the first to suggest running strategy improvements for mean-payoff games or parity games by reduction to discounted games. However Puri's algorithm requires using rational numbers with high precision; each iteration requires solving a linear program involving large numerical values.

Vöge [Vög00] and Vöge and Jurdziński [VJ00] overcame this difficulty for parity games by showing how to run the same algorithm directly on the parity game. In particular, each iteration can be computed in time $O(mn)$, with a purely combinatorial algorithm. On the downside, the required valuation is a bit more involved, and usually only defined over simple lassos. In this scenario, computing the value of a given strategy (or solving a graph), corresponds to computing an optimal (positional) counter strategy, which is guaranteed to exist as a special case of discounted games. For completeness, the necessary axioms in [VJ00] are proven directly for parity games, making for a heavy formalism and tedious proofs. A similar treatment of mean-payoff games obtained by a direct reduction to stochastic games can be found in Friedmann's thesis [Fri11b].

It is interesting to observe that the operators associated to discounted valuations, and therefore also those corresponding to the approach of Vöge and Jurdziński, admit unique fixpoints.

2.2 Energy valuation

We have seen that the energy valuation is a fixpoint valuation and moreover it is co-positional over finite arenas (see Chapter 2). It is not hard to see that it is tractable over graphs, a modified Bellman-Ford algorithm is given in [BFL+08] (see full version) with runtime $O(nm)$; we give no more details here. Therefore, it is fit for strategy improvement. Surprisingly, as far as we are aware, this had not been precisely established before. This is probably due to the fact that the link between mean-payoff and energy games was formalised in [BFL+08; BCD+11], roughly at the same time that Friedmann gave his lower bounds [Fri09], which considerably tamed the excitement around strategy improvement.

A very similar framework was given by Björklund, Sandberg and Vorobyov [BSV04a], who essentially showed that one may run strategy improvements using the energy valuation whenever the arena is modified so that Adam has 0-edges to a *retreat vertex* with just a 0-loop, and moreover, only *admissible strategies* τ are considered, which are positional strategies such that G_τ has only > 0 cycles (one also assumes, without loss of generality, that such a strategy is given to initialise the algorithm). In Schewe's work [Sch08], this second condition is replaced by the assumption (which also does not incur loss of generality) that the game is bipartite, but the so-called *retreat vertex* is still necessary. Luttenberger [Lut08] showed that in the presence of a retreat vertex, the formalisms of [BSV04a; Sch08] and [Vög00; VJ00] actually coincide.

We believe that it is conceptually interesting to know that strategy improvements may be ran with the energy valuation without having to add a retreat vertex, or restrict to admissible strategies. This question was explicitly asked by Björklund, Sandberg and Vorobyov [BSV04a] (see conclusion page 27). In our opinion, this is the simplest available strategy improvement framework for mean-payoff games (or parity games by reduction), and our high-level proof is simply based on their co-positionality over finite arenas, which is well established. We will also use this insight to simplify the algorithm of [Sch08; Lut08] in the next chapter.

An important difference with formalisms based on discounted valuations is that here, fixpoints are not necessarily unique. By reduction, the valuation obtained for parity games corresponds over simple lassos to the vector of *all* occurrences until the last relevant odd occurrence if the lim sup is even, and $\top$ otherwise, naturally ordered lexicographically. This is similar to [Vög00; VJ00], but also different and (we believe) a bit simpler. Friedmann's counterexamples are likely to also apply to this setting, but we would like to verify this more carefully.

2.3 Parity games

The reduction to energy games immediately gives a fixpoint valuation over ω^d, which is also used by Schewe in [Sch08] together with a retreat vertex (these are called *escape games* in [Sch08]). With the formalism of [Vög00; VJ00] this gives two different possibilities for running strategy improvements over parity games, both of which are also applicable to mean-payoff (or energy) games, and subject to Friedmann's exponential lower bounds.

A natural candidate for a strategy improvement specific to parity games is given by Walukiewicz signatures [Wal96], or equivalently Jurdziński's value iteration [Jur00]: the fixpoint valuation over $\omega^{d/2}$ (or equivalently, over arenas of size $\leqslant n$, over $[0, n-1]^{d/2}$) defined by relevant occurrences of odd priorities. However, and surprisingly, this valuation is not positional for Adam; an example is given in Figure 9.2.

The lack of co-positionally also holds when considering valuations inherited from quasipolynomial construction based on trees of height $d/2$. We believe that this fact underlies the non-availability, as of today, of quasipolynomial strategy improvements.

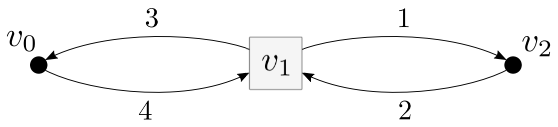


Figure 9.2: A parity game where a non-positional strategy is needed for Adam to achieve the maximal vector of relevant odd occurrences $(1, 1)$ from v_1, by forcing a path of the form $v_1 \xrightarrow{1} v_2 \xrightarrow{2} v_1 \xrightarrow{3} v_0 \xrightarrow{4} \dots$.

As explained in the introduction, designing strategy improvements specific to parity games (be they quasipolynomial or not) is well motivated. We believe that our work opens a very interesting perspective in this direction: can we understand (either in general which might be easier, or specifically for parity games) which monotonic graphs lead to co-positional valuations? What strategy improvement algorithms are induced? These questions should be approached in the light of the one-to-two player lift of Gimbert and Zielonka [GZ05], since fixpoint valuations are positional (for Eve) in general: a fixpoint valuation is co-positional over finite arenas if and only if this is the case for arenas controlled by Adam.

We also mention a recent related work of Koh and Loho [KL21], who present a quasipolynomial strategy improvement algorithm for parity games based on universal trees – seemingly contradicting our discussion. In their framework[1], the progress measure which is computed at each iteration is forced to increase; it does not necessarily correspond to the least fixpoint of G_τ. In particular, each strategy may arise multiple times in the overall iteration, which contrasts with usual strategy iterations (including ours). The main technical contribution of [KL21] is an algorithm establishing a strong graph-tractability property for universal constructions of [JL17], of [DJT20] and based on the complete tree $n^{d/2}$: given a graph G and a progress measure ϕ which is a postfixpoint of Upd_G, one may compute the least prefixpoint which is $\geqslant \phi$.

2.4 Other perspectives

Besides perspectives for parity games, we also believe that our framework could be useful for its flexibility, for simplifying the study of strategy improvement iterations, or extending their applicability. First, could one describe the behaviour of Friedmann's intricate examples in terms of potential reductions (see next chapter), inherent to the energy approach? Could such insights be used to break their mechanics by using other switching policies (designed specifically for energy games)? Or could we at least simplify the examples, and understand better the limits of strategy improvement? (We expect all these questions to be complicated, but worth exploring.)

Second, could the approach be generalised to non-positional (but still structured) Adam strategies? A very interesting case study would be mean-payoff parity games (see Chapter 7), for which designing practical algorithms – which are not available today – is well-motivated by applications in reactive synthesis.

[1] See the Cramer computation frame at the top of page 8 in [KL21].

10 Exploiting symmetry in mean-payoff games

1 Introduction

Deterministic algorithms for energy games. Recently, Dorfman, Kaplan and Zwick [DKZ19] presented the first deterministic algorithm to break the combinatorial $O(n^c 2^n)$ barrier for solving energy games. Their original technique combines a scaling method with an involved subroutine for accelerating the convergence of the BCDGR algorithm. In the final version, the authors realised that scaling can be removed, and the subroutine generalised and simplified to produce an algorithm with complexity

$$O(\min(nmN, m2^{n/2}))$$

for solving energy games. We will call it the DKZ algorithm. It is presented as an acceleration of the one of BCDGR, and inherits their state-of-the-art pseudopolynomial bound. It moreover improves on the state-of-the-art combinatorial bound for deterministic algorithms, which was previously $O(mn2^n)$ from [GKK88; LP07] (see our general introduction for more details).

A careful examination of the DKZ algorithm (obtained after simplification and removal of scaling) reveals a surprising similarity with the GKK algorithm of Gurvich, Karzanov and Khachiyan. However, it is still unclear how to present the GKK algorithm as an acceleration of the one of BCDGR. Moreover, the best known runtime bounds on the GKK algorithm are the pseudopolynomial $O(mn^2N)$ from Pisaruk [Pis99], and the combinatorial $O(mn2^n)$ from the analysis of the original paper [GKK88], both of which are (far) worse than the ones obtained by DKZ.

Contribution and outline. Our main contribution in this chapter is a novel symmetric presentation and analysis of the GKK algorithm when it is ran on a *simple* arena, which are those whose simple cycles have nonzero weight. (This technical assumption can be lifted at the cost of multiplying N by n; more details about it below.) We obtain the pseudopolynomial bound

$$N + E^+ + E^- + 1 \leqslant nN + 1,$$

on the number of its iteration, where E^+ and E^- are respectively the maximum finite energy value and its dual (formally defined below). Each iteration has runtime $O(m)$, therefore this improves (assuming the arena is simple) on the state-of-the-art pseudopolynomial $O(nmN)$ bound.

Moreover, all algorithms so far with runtime $O(nmN)$ are based on value iteration and explicitly involve nN as their maximal finite constant: any instance of a game with vertices of infinite energy values (or equivalently, positive mean-payoff values) has to exceed the value nN, regardless of the structure of the arena. Our bound, although equivalent in the worse case, depends only on energy levels in the arena, which are typically much smaller. We complement our result by adapting the combinatorial analysis of [DKZ19] to the GKK algorithm, improving its combinatorial runtime bound from $mn2^n$ to $m2^{n/2}$, and therefore matching the state of the art.

Section 2 introduces the needed tools, which are (more or less) standard and based on the two natural monotonic graphs for mean-payoff games; we also discuss the simplicity assumption. We present the GKK algorithm in Section 3 and prove the announced pseudopolynomial and exponential bounds in Sections 4 and 5 respectively.

The last section presents a completely independent contribution, which is based on similar tools (introduced in Section 2). We give an alternative (simpler) presentation of the algorithm of Schewe [Sch08] and Luttenberger [Lut08], and propose a symmetric variant which we believe has not been studied so far. More discussion and motivation is given in Section 6.

Related work. We mention a closely related work of Kozachinskiy [Koz21b], who gave an alternative presentation of the DKZ algorithm (also for simple arenas), and extended the framework to discounted games, for which he established the first $2^{O(n)}$ bound, regardless of the discount λ. Kozachinskiy's approach for discounted games can also be presented quite naturally as a variant of the GKK algorithm, but we will not give details here.

Another related work was undergone by Beffara and Vorobyov [BV01], who reported on an empirical study of the GKK algorithm[1], which performs well in practice. They also found that initialising the GKK algorithm with a random potential update (see below) often leads to considerable improvement over their benchmarks; the question is asked whether the randomised algorithm is polynomial over parity games. A surprising recent paper of Lebedev [Leb16] answers in the negative.

Section 6 is based on a joint work with Antonio Casares. We also thank Alexander Kozachinskiy for several interesting and fruitful conversations.

2 Potential reductions and simplicity

Monotonic graphs and symmetry. In contrast with parity games, it appears that mean-payoff games have only very few natural universal monotonic graphs, and this claim is supported by the results of Chapter 6. Informally, we believe that there are only two "meaningful" (or useful) monotonic graphs, which correspond to the usual order over $\mathbb{Z}$ and its dual, and given by

$$\ell \xrightarrow{t} \ell' \qquad \iff \qquad t \leqslant \ell - \ell'$$

or by reversing the order in the right hand side. In the first scenario, a progress measure is a mapping $\phi : V \to \mathbb{Z}$ (or to $\mathbb{Z} \cup \{+\infty\}$) and an edge $v \xrightarrow{t} v'$ is "valid for Eve" if $t \leqslant \phi(v) - \phi(v')$. In the dual point of view, an edge "is valid for Adam" if $t \geqslant \phi(v) - \phi(v')$. By "exploiting the symmetry", we mean "looking at a progress measure from both points of view": an edge can be valid for Eve, for Adam, or for both players (if there is an equality).

Such a symmetry is exploited in the analysis of DKZ, however their algorithm (which follows the framework of BCDGR) is inherently asymmetric. The symmetry is much more apparent in the GKK algorithm, however there is a slight asymmetry which comes from zero cycles. This justifies our assumption of simplicity, with which the GKK algorithm admits a completely symmetric presentation, leading to our improved analysis.

[1] Similar prior studies by Lebedev have also been reported on, but we could not find access to them unfortunately.

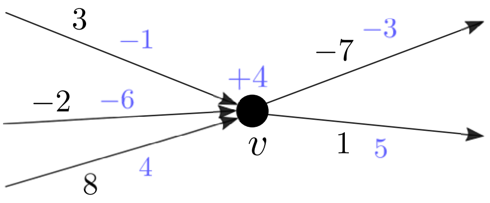


Figure 10.1: Effect of the potential reduction given by $\phi(v) = 4$ and $\phi(v') = 0$ for $v' \neq v$ on the weights of edges adjacent to v. Note that the sum of a path which neither starts nor ends in v is left unchanged.

A very convenient way to approach the study of such progress measures (which we will call *potentials*) is given by potential reductions, which we attribute to [GKK88] in this context and define just below. Potentials are intrinsically linked with energy games, and with the corresponding monotonic graph, and can also be interpreted with the dual point of view. They have numerous occurrences in the literature.

Potential reductions. We fix a finite $\mathbb{Z}$-arena G over V.

A *potential* is a map $\phi : V \to \mathbb{Z}$, which are ordered pointwise. Given an edge $e = v \xrightarrow{t} v'$ in G, we denote

$$t^\phi(e) = t - \phi(v) + \phi(v')$$

which we call the *modified weight* of e. Given e, the edge $e^\phi : v \xrightarrow{t_\phi(e)} v'$ is called the *modified edge*.

The *modified arena* G^ϕ over V is obtained from G by modifying all edges, formally

$$e = v \xrightarrow{t} v' \text{ in } G \qquad \iff \qquad e^\phi = v \xrightarrow{t^\phi(e)} v' \text{ in } G^\phi.$$

A finite or infinite path $\pi = e_0 e_1 \dots$ in G corresponds to a *modified path* $\pi^\phi = e_0^\phi e_1^\phi \dots$ in G^ϕ. Given a finite path $\pi = e_0 \dots e_{n-1} : v_0 \xrightarrow{t_1} \dots \xrightarrow{t_{n-1}} v_n$ of length n, observe that the sum of π^ϕ is related to that of π by the telescopic sum

$$\sum_{i=0}^{n-1} t^\phi(e_i) = -\phi(v_0) + \phi(v_n) + \sum_{i=0}^{n-1} t_i.$$

We call moving from G to G^ϕ a *potential reduction*. The above implies that mean-payoffs of infinite paths, and therefore mean-payoff values of strategies and vertices, are left unchanged by potential reductions, since the constant terms are absorbed in the limit by the multiplication with $1/n$. The following theorem is crucial for us, it relates potential reductions and their effect on energies. It is illustrated in Figure 10.2.

Theorem 10.1 (Potential reductions and energies)

Let ϕ be a potential such that $0 \leqslant \phi \leqslant \mathrm{Energy}_G^+$. Then we have

$$\mathrm{Energy}_{G^\phi}^+ + \phi = \mathrm{Energy}_G^+.$$

An analogous result was used by Hansen, Miltersen and Zwick [HMZ13] (Lemma 3.6) in the setting of discounted games (where the hypothesis vanishes, essentially since there is a unique fixpoint). We prove it as an application of Lemma 1.6 from Chapter 2 but it can of course be proved directly.

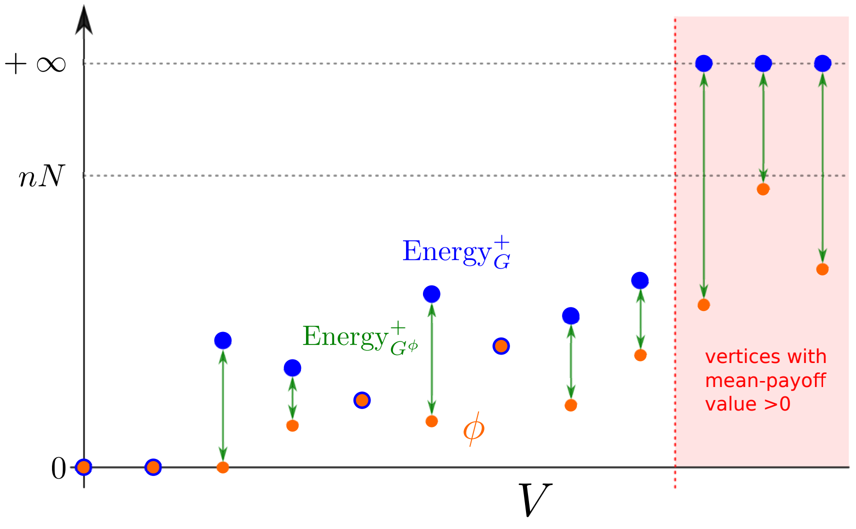

$+\infty$
nN
0
Energy^+_G
$\text{Energy}^+_{G^\phi}$
ϕ
V
vertices with mean-payoff value >0

Figure 10.2: An illustration of Theorem 10.1. For vertices on the right, energy values in both arenas are ∞.

Proof. Consider the construction $\mathcal{L}$ given in Chapter 2 for energy games, and let σ be a uniform positional strategy which respects the prefixpoint $\text{Energy}^+_G : V \to \mathcal{L}^\top$. Let $\pi = e_0 \dots e_{n-1} : v_0 \xrightarrow{t_0} \dots \xrightarrow{t_{n-1}} v_n$ be a finite path in G consistent with σ. By Lemma 1.6 the sum of π satisfies

$$\sum_{i=0}^{n-1} t_i \leqslant \text{Energy}^+_G(v_0) - \text{Energy}^+_G(v_n).$$

Adding $-\phi(v_0) + \phi(v_n)$ on both sides yields

$$\sum_{i=0}^{n-1} t^\phi(i) \leqslant (\text{Energy}^+_G(v_0) - \phi(v_0)) - (\text{Energy}^+_G(v_n) - \phi(v_n)) \leqslant \text{Energy}^+_G - \phi(v_0)$$

since $\phi \leqslant \text{Energy}^+_G$. Therefore we have

$$\phi(v_0) + \sum_{i=0}^{n-1} t^\phi(i) \leqslant \text{Energy}^+_G(v_0),$$

which implies the first inequality.

Conversely, we let σ^ϕ be a uniform positional strategy which respects $\text{Energy}^+_{G^\phi} : V \to \mathcal{L}^\top$. We have for any consistent path π as above that

$$\sum_{i=0}^{n-1} t_i^\phi \leqslant \text{Energy}^+_{G^\phi}(v_0) - \text{Energy}^+_{G^\phi}(v_n),$$

and by adding $\phi(v_0) - \phi(v_n)$ we get

$$\sum_{i=0}^{n-1} t_i \leqslant (\text{Energy}^+_{G^\phi}(v_0) + \phi(v_0)) - (\text{Energy}^+_{G^\phi}(v_n) + \phi(v_n)) \leqslant \text{Energy}^+_{G^\phi}(v_0) + \phi(v_0),$$

since $\phi \geqslant 0$. □

We say that a potential is *positively safe* for G if it satisfies the hypothesis of the theorem

$$0 \leqslant \phi \leqslant \mathrm{Energy}^+_G.$$

In particular, if ϕ coincides with Energy^+_G where it is finite, then the theorem tells us that all vertices have Energy^+-value 0 or ∞ in G_ϕ.

Dual energy. We define the dual-energy valuation over $\mathbb{Z}$ by

$$\mathrm{Energy}^-(t_0 t_1 \dots) = \inf_{n \in \mathbb{N}} \sum_{i=0}^{n-1} t_i \in [-\infty, 0].$$

Note that potentials ϕ and $\phi + c$ where c is a constant define the same reduction. Therefore it is reasonable to consider potentials up to shifts. Since it is convenient to work with non-negative potentials, given a potential ϕ we define

$$\phi^- = \phi - \max \phi$$

which is non-positive in general. We say that a potential ϕ is *negatively safe* if $\mathrm{Energy}^-_G \leqslant \phi^- \leqslant 0$. The dual version of Theorem 10.1 (obtained by reversing the sign of the weights) states that whenever ϕ is negatively safe we have

$$\mathrm{Energy}^-_{G^\phi} + \phi^- = \mathrm{Energy}^-_G.$$

We say that a potential is *bi-safe* if it is both positively and negatively safe, in which case both versions of the theorem can be applied.

Observe that we have $(G^\phi)^{\phi'} = G^{\phi+\phi'}$: sequential applications of potential reductions correspond to reducing with respect to the sum of the potentials.

Lemma 10.1 (Compositionality of safe reductions)

If ϕ is positively (or negatively, or bi-) safe for G and ϕ' is positively (or negatively, or bi-) safe for G^ϕ then $\phi + \phi'$ is positively (or negatively, or bi-) safe for G.

Proof. We first show the result for positively safe. Clearly $\phi + \phi'$ is non-negative since both are. Now Theorem 10.1 gives

$$\mathrm{Energy}^+_G = \mathrm{Energy}^+_{G^\phi} + \phi \geqslant \phi' + \phi,$$

the sought inequality.

For negatively safe, again non-positivity of $(\phi+\phi')^-$ is direct. Using the fact that $\max(\phi+\phi') \leqslant \max(\phi) + \max(\phi')$ in general we obtain similarly

$$\begin{aligned}(\phi + \phi')^- &= \phi + \phi' - \max(\phi + \phi') \geqslant \phi - \max \phi + \phi' - \max \phi' \\ &= \phi^- + \phi'^- = \mathrm{Energy}^-_G - \mathrm{Energy}^-_{G^{\phi^-}} + \phi'^- \geqslant \mathrm{Energy}^-_G.\end{aligned}$$

This also gives compositionality of bi-safety by conjunction. □

This allows for iterative reasoning: if a (positively, negatively, or bi-) safe potential ϕ is found, one may apply the reduction then focus only on finding a potential which is safe for G^ϕ, essentially discarding G.

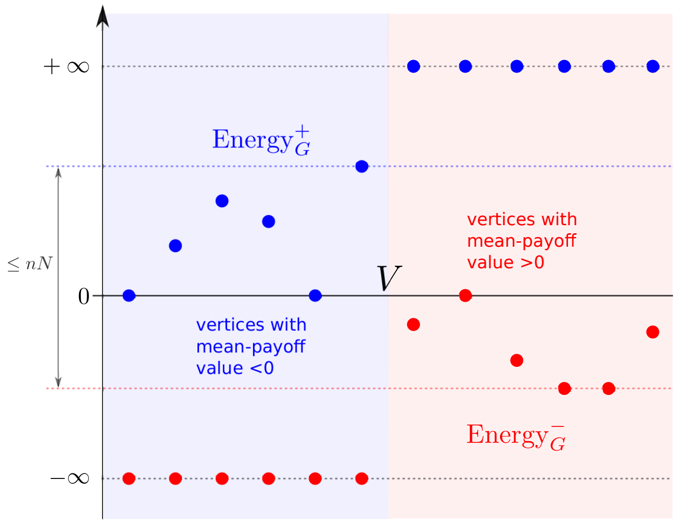


Figure 10.3: Representation of energy values when no vertex has mean-payoff value zero; this is always the case for simple arenas.

Simple arenas. We say that an arena G is *simple* if all simple cycles in G have nonzero sum. In particular a simple arena has no vertex of mean-payoff value zero, since by positionality values coincide with means of simple cycles.

Note that potential reductions preserve the sum of cycles, and therefore if G is simple then so is G^ϕ. We will present the GKK algorithm only over simple arenas for which it admits a description which is completely symmetric.

Given any arena G of size n, let G^+ and G^- denote respectively the arenas obtained by multiplying all weights by n and adding one, and by multiplying all weights by n and subtracting one. It is easy to see that G^+ and G^- are simple, and that for each v we have

$$\begin{array}{lcl} x < 0 & \iff & x^+ < 0 \text{ and } x^- < 0 \\ x = 0 & \iff & x^+ > 0 \text{ and } x^- < 0 \\ x > 0 & \iff & x^+ > 0 \text{ and } x^- > 0 \end{array}$$

where x, x^+ and x^- denote the mean-payoff values of v respectively in G, G^+ and G^-. Therefore we may assume simplicity in general at the cost of blowing up the largest absolute value of a weight N by a multiplicative factor of n.

Note that arenas obtained from the standard reduction from parity games (see preliminaries) are simple. We do not know if mean-payoff or energy games arising from practical applications are typically simple, or if the performing the above reduction would be necessary. Simple mean-payoff games have made several occurrences in the literature, for instance in [Koz21a] or [BSV04a].

Reduced arenas. We say that an arena is *reduced* if the vertices are partitioned into N^* and P^* such that from N^* Eve can ensure to only see non-positive weights and remain in N^*, and symmetrically. Formally,

- vertices in $V_{\text{Eve}} \cap N^*$ have a non-positive edge towards N^*,
- all edges outgoing from vertices in $V_{\text{Adam}} \cap N^*$ are non-positive and toward N^*,
- vertices in $V_{\text{Adam}} \cap P^*$ have a non-negative edge towards P^*,

- all edges outgoing from vertices in $V_{\text{Eve}} \cap P^*$ are non-negative and towards P^*.

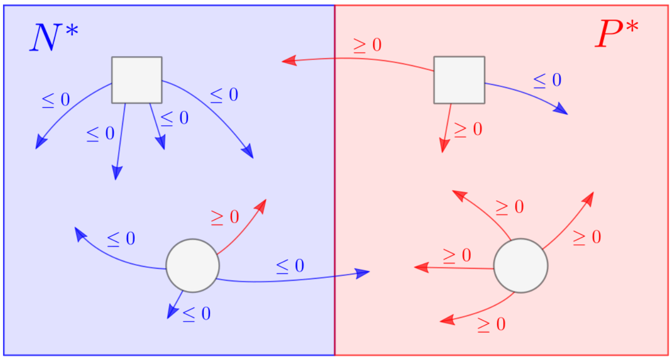


Figure 10.4: A reduced arena. Non-positive edges are represented in blue and non-negative ones in red.

Note that solving a simple reduced arena is trivial: vertices have mean-payoff < 0 if and only if they belong to N^*. In a simple reduced arena, vertices in N^* have Energy$^+$-value 0 and Energy$^-$-value $-\infty$, and symmetrically vertices in P^* have Energy$^-$-value 0 and Energy$^+$-value ∞. (It is not hard to see that this actually characterises simple reduced arenas; we will not use this fact.)

3 Symmetric presentation of the GKK algorithm

We fix a simple arena G. The GKK algorithm iterates potential reductions until obtaining a reduced arena. The runtime for computing each reduction is $O(m)$, therefore the overall runtime is $O(m\ell)$ where ℓ is the number of iterations.

Theorem 10.2 (Number of iterations of GKK over simple arenas)

The number of iterations of the GKK algorithm over simple arenas is bounded by both

$$N + E^+ + E^- + 1 \leqslant nN + 1 \qquad \text{and} \qquad O(2^{n/2}),$$

where E^+ and E^- are respectively the maximal absolute values of finite Energy$^+$ and Energy$^-$ values over V.

Note that we have $E^+ \leqslant \max(n^+ - 1, 0)N$ and $E^- \leqslant \max(n^- - 1, 0)N$, where n^+ and n^- are the respectively number of vertices with positive and negative mean-payoffs values, which satisfy $n^+ + n^- = n$. This implies the inequality on the left.

We now present how each iteration is computed. The two next sections respectively prove the pseudopolynomial and the exponential upper bound.

Description of an iteration. Each iteration relies on a bipartition of the set of vertices, which is completely symmetric thanks to our simplicity assumption. Observe that since there are no simple zero-cycles in G any infinite path visits a non-zero weight. The arena is therefore partitioned into the set of vertices N^* from which Eve can ensure that the first visited non-zero weight is negative, and the set of vertices P^* from which Adam can ensure that the first visited non-zero weight is positive.

Note that the partition N^*, P^* depends only on the signs (and zeroness) of the weights, and not on their precise values. Formally N^* is defined to be the winning region of the objective comprised of all words whose first non-zero weight is negative. It is computable in linear time. In a terminology formally introduced in the next chapter, N^* is the Eve attractor to negative edges over non-positive edges. This justifies our terminology of "attractor-based" for the GKK algorithm.

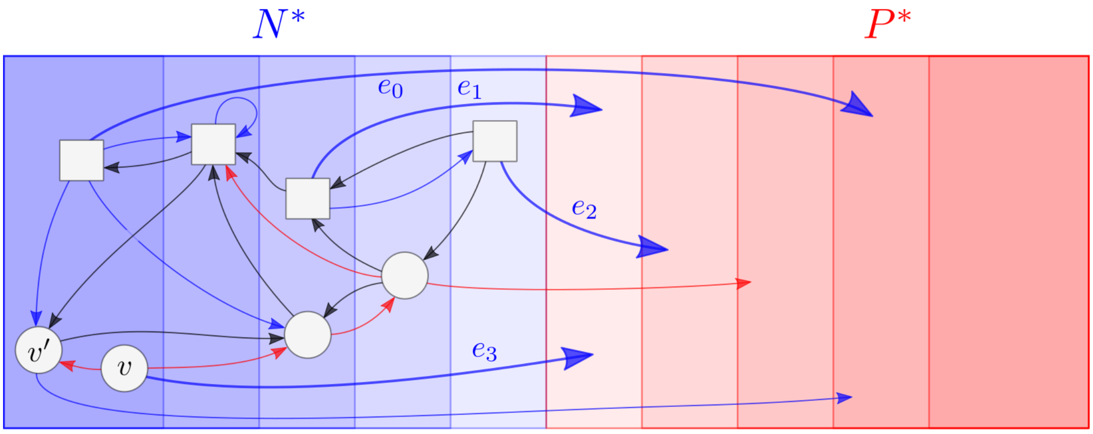


Figure 10.5: An example of the partition of the vertices into N^* and P^*; for clarity, no details are given with respect to P^* where the situation is similar. Blue, black and red arrows respectively represent negative, zero, and positive edges. The layers depicted in N^* correspond to the Eve-attractor over zero edges to negative ones.
With regards to the explanation below: here three edges participate to the maximum defining δ_A^- namely e_0, e_1 and e_2. Only e_3 participates to the maximum defining δ_E^-; v' has a non-positive edge towards N^* and thus does not belong to SN.

As always, we focus on the point of view of Eve, and thus on N^*. By definition from N^* Eve is able to force that a negative edge is seen. The algorithm computes the worst possible (maximal) negative value that Eve can ensure from N^*, which we now describe.

Consider an Adam vertex v in N^*: any edge towards P^* is necessarily negative otherwise v would belong to P^*. Therefore Adam may choose to switch to P^*, but at the cost of seeing a negative weight. We let

$$\delta_A^- = \max\{t \mid N^* \cap V_{\text{Adam}} \xrightarrow{t} P^*\} < 0$$

denote the largest such weight Adam can achieve. It may be that there is no such edge in which case $\delta_A^- = \max \varnothing = -\infty$.

From an Eve vertex v in N^* if Eve has a non-positive edge towards N^* she can follow this path and avoid to switch to P^*. Otherwise all edges outgoing from v towards N^* are positive, and we let

$$SN = \{v \in V_{\text{Eve}} \cap N^* \mid v \xrightarrow{t} N^* \implies t > 0\}$$

be the set of Eve vertices in N^* from which she is forced to switch to P^* or see a positive edge. Note that a vertex $v \in SN$ necessarily has negative outgoing edges, which must therefore point towards P^*, otherwise v would not belong to N^*. Therefore we let

$$\delta_E^- = \max_{v \in SN} \min\{t \mid v \xrightarrow{t} v'\} < 0,$$

and we now put

$$\delta^- = \max(\delta_E^-, \delta_A^-) \in [-\infty, 0).$$

The following result (and the dual one) will be exploited for our pseudopolynomial bound. We prove it now since it refers to the definitions just above.

Lemma 10.2 (Relevance of δ^- in terms of energies)

It holds that Energy_G^- takes values $\leqslant \delta^-$ over N^*.

Proof. Consider a uniform positional strategy σ for Eve which assigns to $v \in (V_{\text{Eve}} \cap N^*) \backslash SN$ a non-positive edge towards N^*, and to $v \in SN$ an edge of weight $< \delta_E^-$ (which therefore necessarily leads to P^*). Consider an infinite path $\pi : v_0 \xrightarrow{t_0} v_1 \xrightarrow{t_1} \dots$ from $v_0 \in N^*$ which is consistent with σ.

If π remains in N^* then all weights are non-positive, and since moreover G is simple it must be that $\text{Energy}^-(\pi) = -\infty$. Otherwise, let $i_0 \in \omega$ be the first index such that $v_{i_0+1} \in P^*$. If $v_{i_0} \in V_{\text{Eve}}$ then necessarily $v_{i_0} \in SN$ and thus $t_{i_0} \leqslant \delta_E^- \leqslant \delta^-$. If $v_{i_0} \in V_{\text{Adam}}$ then likewise $t_{i_0} \leqslant \delta_A^- \leqslant \delta^-$. Since moreover $\pi_{<i_0}$ remains in N^* and is consistent with σ, it only sees non-positive weights, and therefore $\text{Energy}^-(\pi) \leqslant t_0 + t_1 + \dots + t_{i_0} \leqslant \delta$. □

Symmetrically one may define a relevant minimal positive weight for Adam from P^* by setting

$$\delta_E^+ = \min\{t \mid P^* \cap V_{\text{Eve}} \xrightarrow{t} N^*\} \qquad \text{and} \qquad \delta_A^+ = \min_{v \in SP} \max\{t \mid v \xrightarrow{t} v'\}$$

and

$$\delta^+ = \min(\delta_E^+, \delta_A^+) \in (0, \infty],$$

where

$$SP = \{v \in V_{\text{Adam}} \cap P^* \mid v \xrightarrow{t} P^* \implies t < 0\}.$$

We now finally let $\delta = \min(-\delta^-, \delta^+) \in (0, \infty]$. If $\delta = \infty$ then $\delta^- = -\infty$ and $\delta^+ = \infty$ which implies that G is reduced and the iteration stops.

Otherwise we have $\delta > 0$ and we consider the positive potential given by

$$\phi(v) = \begin{cases} \delta & \text{if } v \in P^* \\ 0 & \text{if } v \in N^*. \end{cases}$$

Note that it is symmetric up to shifting by $-\delta/2$, and therefore so is the corresponding potential reduction; it adds δ to the weight of edges from N^* to P^*, removes δ to the weight of edges from P^* to N^*, and leaves other edges unchanged.

Lemma 10.2 implies that ϕ^- is negatively safe, while its dual version (Energy^+ takes values $\geqslant \delta^+ \geqslant \delta$ over G) says that ϕ is positively safe.

4 Pseudopolynomial upper bound

We introduce a terminology from [GKK88]: given a vertex $v \in V_{\text{Eve}}$, its *extremal edges* are its outgoing edges with minimal weight, and extremal edges of $v \in V_{\text{Adam}}$ are its outgoing edges with maximal weight. The *extremal weight* of v is the weight of its extremal edges which we denote $\text{ext}(v) \in \mathbb{Z}$.

We say that a vertex is *negative, zero, or positive*, according to the sign of its extremal weight. We let[2] N, Z and P denote the sets of negative, zero, and positive vertices in G. Note that $N \subseteq N^*$ and $P \subseteq P^*$ while Z is split between both.

[2] We apologise for the clash in notations with our notation N for the maximal absolute value of a weight, which is easily resolved thanks to context.

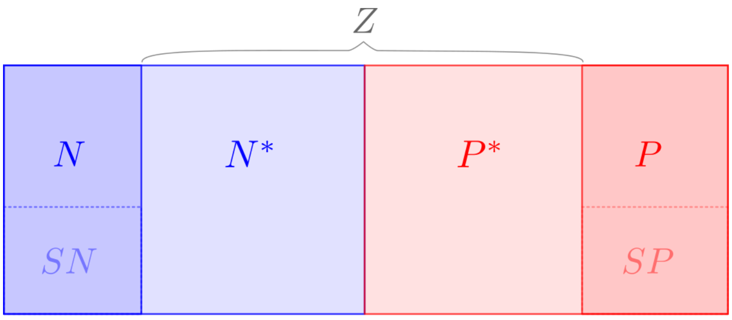


Figure 10.6: The three sets of vertices N,Z and P, in relationship with N^*, P^*. Lemma 10.3 states that from an iteration to the next, no new vertex becomes negative or positive. We also display SN and SP although these are not used for the pseudopolynomial bound (besides relying on Lemma 10.2).

We let $G' = G^\phi$ be obtained from G after the potential reduction introduced above, and use primes to denote subsets of vertices and quantities defined as above in G'.

Lemma 10.3 (Evolution of signs of vertices)

We have

$$Z \subseteq Z', \qquad N \supseteq N' \quad \text{and} \quad P \supseteq P'.$$

Proof. We prove that

$$\begin{aligned} \forall v \in N^*, \quad & \text{ext}(v) \leqslant \text{ext}'(v) \leqslant 0 \\ \forall v \in P^*, \quad & \text{ext}(v) \geqslant \text{ext}'(v) \geqslant 0. \end{aligned}$$

This implies the lemma: if $\text{ext}(v) = 0$ then so does $\text{ext}'(v)$ and if $\text{ext}'(v) < 0$ then necessarily $\text{ext}(v) < 0$. We only prove the first line since the second follows by symmetry.

For the left inequality it suffices to observe that the weight of edges outgoing from N^* can only increase: edges pointing to N^* keep the same weight while those pointing towards P^* are increased by δ. For the inequality on the right we make a quick case disjunction.

- Let $v \in N^* \cap V_{\text{Adam}}$. Then all extremal edges are non-positive, and those which point towards P^* are even $\leqslant -\delta$ by definition of δ hence they all remain non-positive.
- Let $v \in N^* \cap V_{\text{Eve}}$. The result follows directly if v has a non-positive outgoing edge towards N^* since it is left unchanged. Otherwise $v \in SN$ hence v has an outgoing edge of weight $\leqslant -\delta$ which therefore remains non-positive. □

We now let $G = G^0, G^1, G^2 \dots$ denote the sequence of arenas encountered throughout the iteration, and use obvious notations such as N^j, $P^{j,*}$, δ^j or ϕ^j. In particular G^{j+1} is defined if and only if $\delta^j < \infty$.

Given j_0 such that G^{j_0} is defined, we moreover let

$$\Delta^{j_0} = \sum_{j=0}^{j_0} \delta^j \qquad \text{and} \qquad \Phi^{j_0} = \sum_{j=0}^{j_0} \phi^j.$$

The following is a crucial consequence of Lemma 10.3.

Corollary 10.1

It holds that Φ^j takes value 0 over N^j and Δ^j over P^j.

In particular we have $\min \Phi^j = 0$ and $\max \Phi^j = \Delta^j$.

Proof. Thanks to lemma 10.3 we have

$$N^0 \supseteq N^1 \supseteq \cdots \supseteq N^j$$

therefore if $v \in N^j$ then for all $j' \leqslant j$, v belongs to $N^{j'} \subseteq N^{j',*}$ therefore $\phi^{j'}(v) = 0$ and thus $\Phi^j(v) = 0$. Likewise, if $v \in P^j$ then for all $j' \leqslant j$ we have $\phi^{j'}(v) = \delta^{j'}$ therefore $\Phi^j(v) = \Delta^j$. □

With this is hands we are ready to prove the announced bound.

Theorem 10.3 (Novel pseudopolynomial bound)

The iteration terminates in at most $N + E^+ + E^- + 1$ steps.

The proof is illustrated in Figure 10.7.

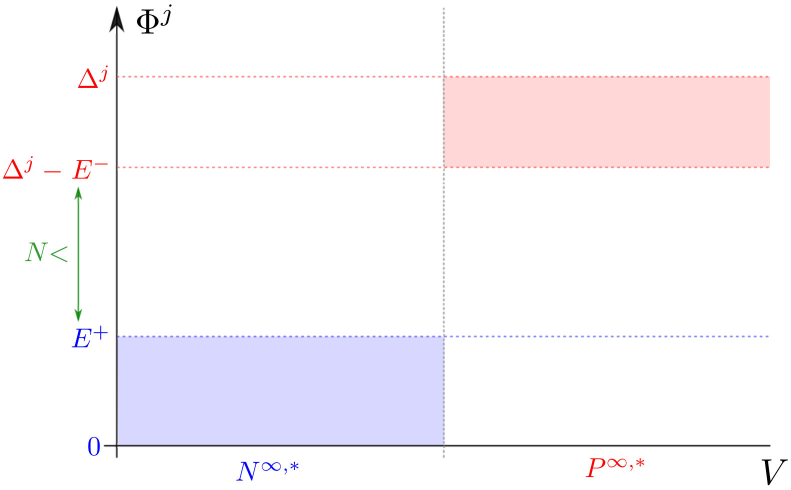


Figure 10.7: An illustration for the proof of Theorem 10.3, where $j = N + E^+ + E^-$. Since Ψ^j it is positively safe, vertices with finite Energy$^+$ value (denoted $N^{\infty,*}$) must be mapped to the blue region, and symmetrically; by our choice of j, this implies that edges from $N^{\infty,*}$ to $P^{\infty,*}$ are positive, and those from $P^{\infty,*}$ to $N^{\infty,*}$ are negative.

Proof. We let $N^{\infty,*}$ and $P^{\infty,*}$ respectively denote the sets of vertices with negative and positive mean-payoff values, which partition G. Since it is positively safe by composition, and the quantities below are finite, we have thanks to Theorem 10.1 for all j that over $v \in N^{\infty,*}$,

$$\Phi^j(v) = \text{Energy}^+_G(v) - \text{Energy}^+_{G^j}(v) \leqslant E^+.$$

Likewise, over $v \in P^{\infty,*}$ we obtain

$$\Phi^{j,-}(v) = \text{Energy}^-_G(v) - \text{Energy}^+_{G^j}(v) \geqslant -E^-,$$

which rewrites as

$$\Phi^j(v) \geqslant \Delta^j - E^-.$$

We now assume that the $j = N + E^+ + E^-$-th iteration is defined, and for contradiction that $\delta^j < \infty$. Note that $\Delta^j \geqslant j + 1$ as a sum of $j + 1$ positive integers.

We claim that N^j (and symmetrically, P^j) is non-empty. Indeed if $N^j = \varnothing$ then $P^j = V$ therefore $\delta^j = \infty$. (Intuitively, Adam can ensure that no negative weight is ever seen.)

By Corollary 10.1, Φ^j takes value 0 over N^j therefore $N^j \subseteq N^{\infty,*}$ thanks to the above since $0 < \Delta^j - E^-$ (see Figure 10.7). Likewise, we have $P^j \subseteq P^{\infty,*}$ since $\Delta^j > E^+$.

Note that any edge $v \xrightarrow{t} v'$ from $N^{\infty,*}$ to $P^{\infty,*}$ has weight

$$t^{\Phi^j}(v) = t + \Phi^j(v') - \Phi^j(v) \geqslant t + \Delta^j - E^- - E^+ \geqslant -N + \Delta^j - E^- - E^+ \geqslant 1$$

in G^j. Likewise, any edge from $P^{\infty,*}$ to $N^{\infty,*}$ has weight < 0 in $\mathcal{G}^j$, therefore zero edges cannot lead from $N^{\infty,*}$ to $P^{\infty,*}$ or vice-versa.

Now note that by definition vertices in $N^{j,*}$ have a path to $N^j \subseteq N^{\infty,*}$ comprised of only zero weights in G^j, and therefore $N^{j,*} \subseteq N^{\infty,*}$; similarly, $P^{j,*} \subseteq P^{\infty,*}$. Therefore we have

$$N^{j,*} = N^{\infty,*} \qquad \text{and} \qquad P^{j,*} = P^{\infty,*}.$$

Since all edges from $N^{j,*}$ to $P^{j,*}$ are positive, we have $\delta^- = -\infty$. Likewise $\delta^+ = \infty$ and therefore $\delta = \infty$, a contradiction. □

5 Strong exponential upper bound

We now prove the $O(2^{n/2})$ bound by adapting the argument of [DKZ19]. We first discuss the general strategy which we break in two steps and then present the two steps separately.

Proof structure. Consider the attractor layers[3] $L_1, L_2 \cdots \subseteq N^*$ towards N over zero edges. It was already observed in [GKK88] that the sequence

$$|L_1 \cap V_{\text{Eve}}|, -|L_1 \cap V_{\text{Adam}}|, |L_2 \cap V_{\text{Eve}}|, -|L_2 \cap V_{\text{Adam}}|, \dots$$

strictly increases lexicographically at each step of the iteration such that $N = N'$ and $P = P'$. Using Lemma 10.3 this yields a $O(n2^n)$ upper-bound on the number of iterations, which is not explicitely given in [GKK88] although the argument is laid out to prove termination (it is also very easy to reduce the bound to $O(2^n)$, directly from their analysis). We follow the strategy of [DKZ19], which we break in two steps.

- The first step consists in exhibiting a different sequence of layers with a similar behaviour. Proving strict lexicographical increase is quite involved (more than for the attractor layers of [GKK88], we believe).
- The second step relies on encoding the above sequence in a strictly growing integer over $\leqslant |N^*|$ bits. Exploiting the symmetry then allows to conclude via an elegant padding argument and lower the bound to $O(2^{n/2})$.

[3] This is formally defined in the next chapter. Here, we discuss them only informally. At this stage, we believe that the reader is likely to be familiar with this standard concept.

It is crucial for the padding argument to apply that the nonzero signs in the above sequence alternate between positive and negative, which might not be the case with the layers as described above (for instance it might be that the second integer is zero, while the first and third are positive). One may circumvent this issue by reducing (without loss of generality) to a bipartite arena which ensures the wanted property and allows to use the padding argument (this is done in [Koz21b]) directly on the attractor layers, invoking the proof of [GKK88].

We believe however that the layers of [DKZ19] together with the proof of step one are of independent interest, hence we will follow their definitions and directly adapt their argument to the case at hands. As we will see, the layers of [DKZ19] are defined only via *paths*, and thus appear to be less natural than the attractor layers of [GKK88] in this setting. It is therefore quite surprising to us that the main result (Theorem 10.4) still holds. Besides, we do not know if a similar padding argument can be forged directly for the attractor layers (because there might be zeros, as explained above), therefore using the layers of [DKZ19] seems necessary to establish the result over non-bipartite arenas.

5.1 Step one: layers and their dynamics

Again, we focus on N^*, but will later use the main result together with its dual to obtain the wanted bound. Given a finite path $\pi : v_0 \to v_1 \to \dots \to v_\ell$ in G we define its number of *alternations (towards N)* $\mathrm{alt}(\pi) \in \omega \cup \{\infty\}$ to be the minimal k such that there exist a decreasing sequence of $k+1$ indices $\ell \geqslant i_0 \geqslant i_1 \geqslant \dots \geqslant i_k$ such that

- $v_{i_0}, \dots, v_\ell \in N$,
- for all $j \in [1,k]$, $v_{i_j}, \dots, v_{i_{j-1}-1}$ all belong to V_{Adam} if j is odd and to V_{Eve} if j is even.

In particular a path has finite alternation number if and only if it ends in N and it has alternation number 0 if and only if it is contained N. Moreover note that a path from $v \notin N$ towards N has even alternation number if and only if $v \in V_{\mathrm{Eve}}$. The choice of the first layer being comprised of Adam vertices is arbitrary, the proof below also goes through with the other convention.

We say that a path is *zero* if it visits only zero edges. We define the *alternation depth* $\mathrm{alt}(v)$ over vertices in N^* by

$$\mathrm{alt}(v) = \min\{\mathrm{alt}(\pi) \mid \pi \text{ is a zero path from } v \text{ to } N \text{ which remains in } N^*\}.$$

An example is given in Figure 10.8. We say that a path from $v \in N^*$ is *optimal* if it is a zero path from v to N which remains in N^* and achieves the above minimum. Note that by definition of N^*, vertices in N^* have a simple zero path towards N hence $\mathrm{alt}(v)$ is finite and bounded by n.

We will study the dynamics of the sets

$$A_i = \{v \in N^* \mid \mathrm{alt}(v) = i\}.$$

We assume that the iteration is not over, $\delta < \infty$. We again use the notation G' for G^ϕ, where ϕ is the GKK potential (defined in Section 3), and again use primes for sets and quantities relative to G'. The following is the main result for the first step.

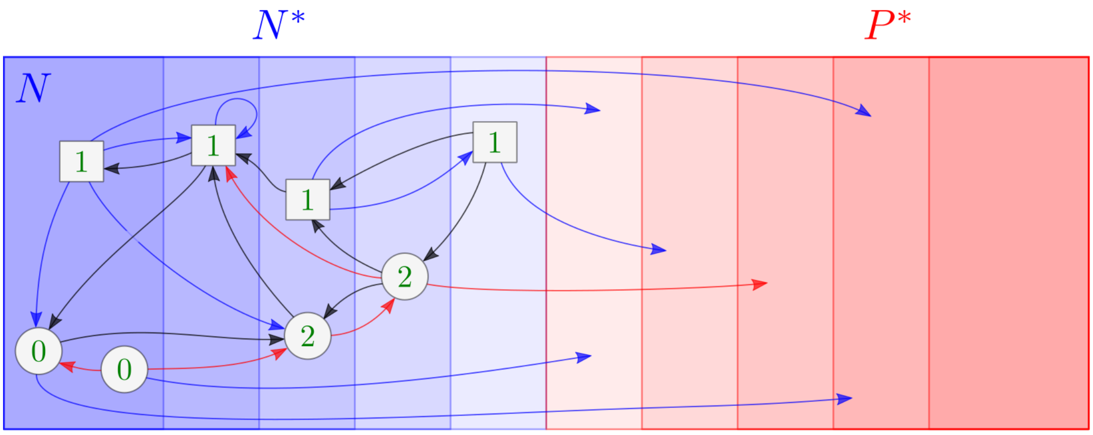


Figure 10.8: The alternating layers, indicated by the green numbers, in the example of Figure 10.5. Notice that alternating layers (green numbers) and attractor layers (in blue) are completely different; however – and quite surprisingly – the theorem below holds in both cases (see [GKK88] for details about attractor layers).

Theorem 10.4 (Dynamics of alternating layers)

If $N = N'$ and $P = P'$, then the sequence

$$-|A_1|, |A_2|, -|A_3|, |A_4|, \dots$$

stricly grows lexicographically.

Towards proving the theorem, we define two relevant indices i_D and i_A which we respectively call the *departure index* and *arrival index*. As their names suggest the first is relevant to vertices which leave N^*, that is, those in $N^* \cap P'^*$, while the second is relevant to arriving vertices, those in $P^* \cap N'^*$. We let

$$\begin{array}{rcl} i_D & = & \min\{\mathrm{alt}(v) \mid v \in V_{\text{Adam}} \cap N^* \text{ and } v \in P'^*\}, \\ i_A & = & \min\{\mathrm{alt}(e_0\pi_1) \mid e_0 \text{ is an edge of weight } \delta \text{ from } P^* \cap V_{\text{Eve}} \text{ to } v_1 \in N^* \text{ in } G \\ & & \hfill \text{and } \pi_1 \text{ is optimal from } v_1 \text{ in } G\} \end{array}$$

Note that if finite, i_D is odd and i_A is even. We now provide a sequence of incremental results that eventually give the theorem.

Lemma 10.4 (Incremental statements proving Theorem 10.4)

Assume that $N' = N$ and $P' = P$.

(i) For all $v \in N^*$, if $\mathrm{alt}(v) < i_D$ then $v \in N'^*$ and $\mathrm{alt}'(v) \leqslant \mathrm{alt}(v)$.

(ii) Any path π' which is optimal in G' but is not zero in G satisfies $\mathrm{alt}'(\pi') \geqslant i_A$.

(iii) For all $v \in P^* \cap N'^*$ it holds that $\mathrm{alt}'(v) \geqslant i_A$.

(iv) For all $i \leqslant \min(i_D - 1, i_A)$ we have $A_i \subseteq A'_i$ and for all $i \leqslant \min(i_D, i_A - 1)$ we have $A'_i \subseteq A_i$.

(v) If $i_A < i_D$ then $|A'_{i_A}| > |A_{i_A}|$.

(vi) We have $\delta^-_E < -\delta$ and likewise $\delta^+_A > \delta$.

(vii) If $i_A = \infty$ then $i_D < \infty$.

(viii) Theorem 10.4 holds.

Items (i), (ii) and (iii) build towards item (iv) which is the main intermediate result. Items (v) and (vii) have a similar proof although (vii) also relies on (vi), and build up to the conclusion.

Proof. (i) We prove the claim by induction on the length ℓ of the smallest optimal path $\pi = v_0 \xrightarrow{0} \dots \xrightarrow{0} v_\ell$ from $v_0 = v$. Note that π is zero in G and remains in N^* hence it is also zero in G'. If π has length zero then $v \in N = N'$ hence $v \in N'^*$ and $\mathrm{alt}'(v) = 0 \leqslant \mathrm{alt}(v)$, so we assume $\ell > 0$ and that the result is known for vertices with an optimal path of length $\leqslant \ell - 1$.

It holds by induction that $v_1, v_2, \dots, v_\ell \in N'^*$ hence it suffices to prove that $v \in N'^*$ since it implies that π is a zero path in G' which remains in N'^*. If $v \in V_{\text{Eve}}$ then v has a zero edge in G' towards $v' \in N'^*$ hence $v \in N'^*$. Otherwise it holds that $v \in N'^*$ because $v \in P'^*$ would contradict that $\mathrm{alt}(v) < i_D$.

(ii) Let $\pi' : v_0 \xrightarrow{t_0} \dots \xrightarrow{t_{\ell-1}} v_\ell$ be such a path. It cannot be that π' is included in N^* otherwise it would be zero in G, and we let i_0 be the largest index such that $v_{i_0} \in P^*$. Since $t'_{i_0} = 0$ we have $t_{i_0} = \delta > 0$ hence it must be that $v_{i_0} \in V_{\text{Eve}}$ otherwise we would have $v_{i_0} \in P = P'$ which contradicts that $v_{i_0} \in N'^*$. We now let π be an optimal path from v_{i_0+1}. Then we have

$$\mathrm{alt}(\pi') \geqslant \mathrm{alt}((v_{i_0} \xrightarrow{t_{i_0}} v_{i_0+1})\pi) \geqslant i_A.$$

(iii) Let $v \in P^* \cap N'^*$, and let $\pi' : v_0 \xrightarrow{t_0} \dots \xrightarrow{t_{\ell-1}} v_\ell$ be an optimal path from $v = v_0$ in G'. We assume for contradiction that $\mathrm{alt}'(v) < i_A$, which thanks to the previous item implies that π' is zero in G. Since $v_\ell \in N' = N$ and $v = v_0 \in P^*$ there is an index i_0 such that $v_{i_0} \in P^*$ and $v_{i_0+1} \in N^*$. This contradicts the fact that $t_{i_0} = t'_{i_0} = 0$.

(iv) We prove the two results together by induction on i. For $i = 0$ we have $A_0 = N = N' = A'_0$ hence we let $i \geqslant 1$ and assume that both result hence the equality are known for smaller values.

By item (i) if $i < i_D$ and $v \in A_i$ then $v \in N'^*$ and $\mathrm{alt}'(v) \leqslant i$, but our induction hypothesis tells us that $\mathrm{alt}'(v)$ cannot be $< i$ hence $A_i \subseteq A'_i$.

Conversely let $v \in A'_i$, assume $i < i_A$, let $\pi : v_0 \xrightarrow{t_0} \dots \xrightarrow{t_{\ell-1}} v_\ell$ be an optimal path from $v_0 = v$ in G', and let $j_0 > 0$ be the smallest index such that $v_{j_0} \notin A'_i$. We assume that π is chosen such that v_{j_0} is minimal, and prove the result by an inner induction on j_0. Since $\mathrm{alt}'(v) = i < i_A$ we know by item (iii) that $v \in N^*$.

If $j_0 = 1$, that is if $v_1 \in A'_{i-1}$, then thanks to the (outer) induction hypothesis for all $j \geqslant 1$ we have $v_j \in A'_{k_j} = A_{k_j}$ for some $k_j < i$, hence for all $j \geqslant 0$ we have $v_j \in N^*$. Hence π' remains in N^* and is zero in G' thus it is also zero in G and $\mathrm{alt}(v) \leqslant \mathrm{alt}(\pi) = i$. We conclude thanks to the (outer) induction that $\mathrm{alt}(v) = i$.

If $j_0 \geqslant 2$ then the inner induction hypothesis gives $v_j \in N^*$ for $j \in [1, j_0]$ and the outer induction hypothesis gives $v_j \in N^*$ for $j \in [j_0 + 1, \ell]$, and we repeat the same argument.

(v) Assume that $i_A < i_D$. By item (iv) it holds that $A_{i_A} \subseteq A'_{i_A}$ hence it suffices to find $v_0 \in A'_{i_A} \backslash A_{i_A}$ and we take v_0 given by the definition of i_A: $v_0 \xrightarrow{t_0} v_1$ with $v_0 \in P^* \cap V_{\text{Eve}}$, $v_1 \in N^*$ with $\mathrm{alt}(v_1) \leqslant i_A$ and $t_0 = \delta$ therefore $t'_0 = 0$.

Again by (iv) it holds that $v_1 \in N'^*$ and $\mathrm{alt}'(v_1) \leqslant \mathrm{alt}(v_1)$, hence $v_0 \in V_{\text{Eve}}$ has a zero edge in G' towards a vertex of N'^* thus $v_0 \in N'^*$. Now $\mathrm{alt}'(v_0) \leqslant \mathrm{alt}((v_0 \xrightarrow{t'_0} v_1)\pi'_1)$, where π'_1 is an optimal path from v_1 in G' and hence $\mathrm{alt}'(v_0) \leqslant i_A$. Yet again thanks to (iv) it cannot be that $\mathrm{alt}'(v_0) < i_A$ since $A'_i \subseteq A_i$ for $i < i_A$ and $v_0 \in P^*$, therefore we conclude that $v_0 \in A'_i$.

(vi) Assume for contradiction that $\delta_{N,2} = -\delta$. Then there is $v \in SN$ such that $\mathrm{ext}(v) = -\delta$, hence $\mathrm{ext}'(v) = 0$ so $v \in Z'$, which contradicts $N' = N$. The proof of the second statement is symmetric.

(vii) If $i_A = \infty$ then there is no edge with weight δ in G from $P^* \cap V_{\text{Eve}}$ to N^* hence $\delta_{P,1} > \delta$ therefore it must be by item (vi) that $\delta = -\delta^-_A$. We let $e_0 = v_0 \xrightarrow{t}_0 v_1$ be an edge with weight $t_0 = -\delta$ from $v_0 \in V_{\text{Adam}} \cap N^*$ to $v_1 \in P^*$. We claim that $P'^* \supseteq P^*$ which proves the result since then $v_0 \in V_{\text{Adam}}$ has an edge e_0 which is zero (hence non-negative) in G' towards P'^*, hence $v_0 \in P'^*$ and $i_D \leqslant \mathrm{alt}(v_0)$.

This follows from a quick induction over attractor-layers towards $P = P'$ over zero edges in G: a vertex $v \in V_{\text{Adam}} \cap (P^* \backslash P)$ has a zero edge, which remains zero, in G towards a vertex in the previous layer, and by assumption vertices $v \in V_{\text{Eve}} \cap (P^* \backslash P)$ have all their edges towards N^* which are $\geqslant \delta_{P,1} > \delta$ hence remain positive.

(viii) By item (vii) $m = \min(i_D, i_A)$ is finite, and by item (iv) we have $A_i = A'_i$ for all $i < m$. If $m = i_A \leqslant i_D - 1$ then moreover $A_m \subseteq A'_m$ and the inclusion is strict by item (v), which concludes. Otherwise $m = i_D \leqslant i_A - 1$ hence $A_m \supseteq A'_m$ and the inclusion is strict by definition of i_D. □

Even broken in elementary steps the proof above remains very tedious, we are not aware unfortunately of simplifications that could be made. The remainder is much more straightforward.

5.2 Step two: padding argument

We now describe the second step of the upper-bound proof, which is due to [DKZ19]. We let k denote $|P| + |N|$ which can only decrease throughout the iteration thanks to Lemma 10.3. Note that there exists $r \in [1, n-k]$ such that the layers $A_1, \dots, A_r$ are non-empty, and $A_{r+1}, A_{r+2}, \dots$ are empty. We let $s_r = 1$ if r is even and 0 otherwise.

The argument relies on the following $n - k + 1$-bits integer

$$\alpha^- = \underbrace{0\dots0}_{|A_1|}\,\underbrace{1\dots1}_{|A_2|}\,\underbrace{0\dots0}_{|A_3|}\dots\underbrace{s_r\dots s_r}_{|A_r|}\,1\,\underbrace{0\dots0}_{|P^*|-|P|},$$

and its symmetric counterpart α^+, which is defined in exactly the same way with respect to layers in P^*.

Lemma 10.5 (Quantifying the growth of α_N)

If $k = k'$ then $\alpha'^- > \alpha^- + 2^{|P^*|-|P|}$ and likewise $\alpha'^+ > \alpha^+ + 2^{|N^*|-|N|}$.

Proof. By Theorem 10.4 the leftmost bit to switch from α^- to α'^- switches from 0 to 1, and occurs before the rightmost block of the form $10\dots0$ with $|P^*| - |P|$ zeros. □

We are finally ready to prove the announced bound.

Proof of $O(2^{n/2})$ bound. Consider $\alpha = \alpha^- + \alpha^+$, which is $\leqslant 2^{n-k+2}$. Note that $|N^*| - |N| + |P^*| - |P| = n - k$, hence $\max(|N^*| - |N|, |P^*| - |P|) \geqslant \frac{n-k}{2}$. By the above lemma, if $k' = k$ then

$$\alpha' > 2^{\max(|N^*|-|N|,|P^*|-|P|)} \geqslant 2^{\frac{n-k}{2}}.$$

Hence, there are at most $2^{n-k+2}/2^{\frac{n-k}{2}} = 4.2^{\frac{n-k}{2}}$ consecutive iterations with the same k. The bound follows since

$$\sum_{k=0}^{n-1} 4.2^{\frac{n-k}{2}} = O(2^{n/2}). \qquad \square$$

6 The ESL algorithm

In this independent section, we discuss an algorithm due to Schewe [Sch08], which was also presented in a different light by Luttenberger [Lut08].

Informal presentation. Schewe presents it as a switching policy, roughly in the strategy improvement framework of Björklund and Vorobyov [BV05] discussed in the previous chapter, with a retreat vertex. In this setting, it is *locally optimal* in the sense that at each iteration, a combination of switches is performed to obtain the strategy τ' maximising $\text{val}(G_{\tau'})$ among all strategies available by combining improving switches. Moreover, the iteration can be computed in almost linear $O(m + n \log n)$ time. Schewe observed that the algorithm consistently outperforms (often, by far) all other switching policy over practical and random instances, which makes it to this date the most efficient and reliable strategy improvement algorithm.

Luttenberger [Lut08] gave a completely different presentation of Schewe's algorithm, as one improving over *non-deterministic* strategies, still with a retreat vertex and a restriction over admissible strategies (called reasonable strategies in [Lut08]). In the setting of non-deterministic strategy improvement which he introduces, the algorithm rephrases naturally as the "all profitable switches" policy: at each step, the (non-deterministic) strategy τ' is simply comprised of all improving switches. Luttenberger complements his results by showing that the strategy improvement frameworks of [BV05] and [Vög00; VJ00] coincide when the arena has a retreat vertex.

Our contribution. Following our result in the previous chapter that the energy valuation is fit for strategy improvement, we investigate whether this algorithm can be understood in the vocabulary of energy games (without retreat vertices or strategy restrictions). It turns out that it has a surprisingly natural presentation in this setting, and we like to see it as an acceleration of the value iteration of [BCD+11] rather than a strategy improvement. We will call the resulting algorithm the ESL algorithm (for Energy-Schewe-Luttenberger). We recall that Luttenberger's version is implemented (over GPUs) in STRIX [MSL18; LMS20], the currently most efficient academic tool for LTL synthesis. We believe that our approach is more manageable and flexible, and might be prone to more efficient implementations.

Just like the BCDGR algorithm, the ESL algorithm is completely asymmetric. We also propose a natural symmetric variant, which had not been considered before. Quite surprisingly, we are not able to prove its termination, although our simulations suggest that the algorithm not only terminates, but performs even fewer iterations over random instances, especially those induced by parity games. We first present the ESL algorithm, then its symmetric variant, and then discuss a few early empirical results.

6.1 Presentation of the algorithm

Thanks to Theorem 10.1, we may consider algorithms which iterate positively safe potential reductions, formally, given by ϕ satisfying

$$0 \leqslant \phi \leqslant \mathrm{Energy}^+_G.$$

The GKK algorithm is of this form (see Section 3), and stops when the obtained arena is reduced. The BCDGR algorithm is also of this form, where $\phi(v)$ is defined to be the extremal weight of v if v is positive (or invalid), and 0 otherwise; moreover an infinite potential is given to vertices which exceed nN, to ensure termination. Formalising the BCDGR algorithm in terms of potential reductions therefore requires introducing potentials with infinite coordinates, which is done just below. One can also see the DKZ algorithm (whose details we will omit) in this scenario: roughly speaking, the potential reduction which is performed is similar to GKK's, while precise mechanics which guarantee the $O(nmN)$ implementation are inspired from BCDGR's.

The ESL algorithm is also given in this form, where ϕ is the maximal weight of a path comprised of only non-negative weights which Adam can ensure. Again, it is convenient to present the algorithm only over simple arenas. A complete execution is depicted in Figure 10.9.

Sum until negative valuation. We extend the energy valuation to $\mathbb{Z}^\infty = \mathbb{Z} \cup \{\infty\}$ by setting as previously

$$\begin{array}{rrcl} \mathrm{Energy}^+ : & (\mathbb{Z}^\infty)^\omega & \to & [0,\infty] \\ & t_0 t_1 \dots & \mapsto & \sup_k \sum_{i=0}^{k-1} t_i \end{array}$$

where naturally $\infty + t = \infty$ for any $t \in \mathbb{Z}^\infty$. We now define the *sum until negative* valuation SU^+ by

$$\begin{array}{rrcl} \mathrm{SU}^+ : & (\mathbb{Z}^\infty)^\omega & \to & [0,\infty] \\ & t_0 t_1 \dots & \mapsto & \sum_{i=0}^{k_{\mathrm{neg}}-1} t_i \quad \text{where } k_{\mathrm{neg}} = \min\{k \mid t_k < 0\}. \end{array}$$

It follows directly from the definition that for any $w = t_0 t_1 \dots \in (\mathbb{Z}^\infty)^\omega$, we have $0 \leqslant \mathrm{SU}^+(w) \leqslant \mathrm{Energy}^+(w)$. Therefore, SU^+_G defines a positively safe potential. Note also that SU^+_G is larger in general than the potential of the BCDGR algorithm, which only considers the first letter (if it is positive).

Extended potentials. We fix a $\mathbb{Z}^\infty$-arena G. An *extended potential* is a map $\phi : V \to \mathbb{Z}^\infty$. The modified weight of an edge $e = v \xrightarrow{t} v'$ is now given by

$$t^\phi(e) = \begin{cases} \infty & \text{if } t, \phi(v) \text{ or } \phi(v') \text{ is } \infty \\ t - \phi(v) + \phi(v') & \text{otherwise.} \end{cases}$$

This definition may seem odd at first sight; the intuition is that vertices mapped to ∞ are automatically declared winning for Adam (meaning, of infinite energy, or positive mean-payoff). Modified arenas, modified paths, and potential reductions are defined just like previously. We extend Theorem 10.1 to extended potentials, which is just a formality. The statement is exactly the same.

[4] We have not established positionality of SU^+, which holds over arbitrary arenas. One may easily prove this result by showing that the completely well-monotonic graph over $\omega \cup \{\top\}$ given by $\ell \xrightarrow{t} \ell' \iff (\ell = \top \text{ or } 0 \leqslant t \leqslant \ell - \ell' \text{ or } t < 0)$ is uniformly SU^+-universal.

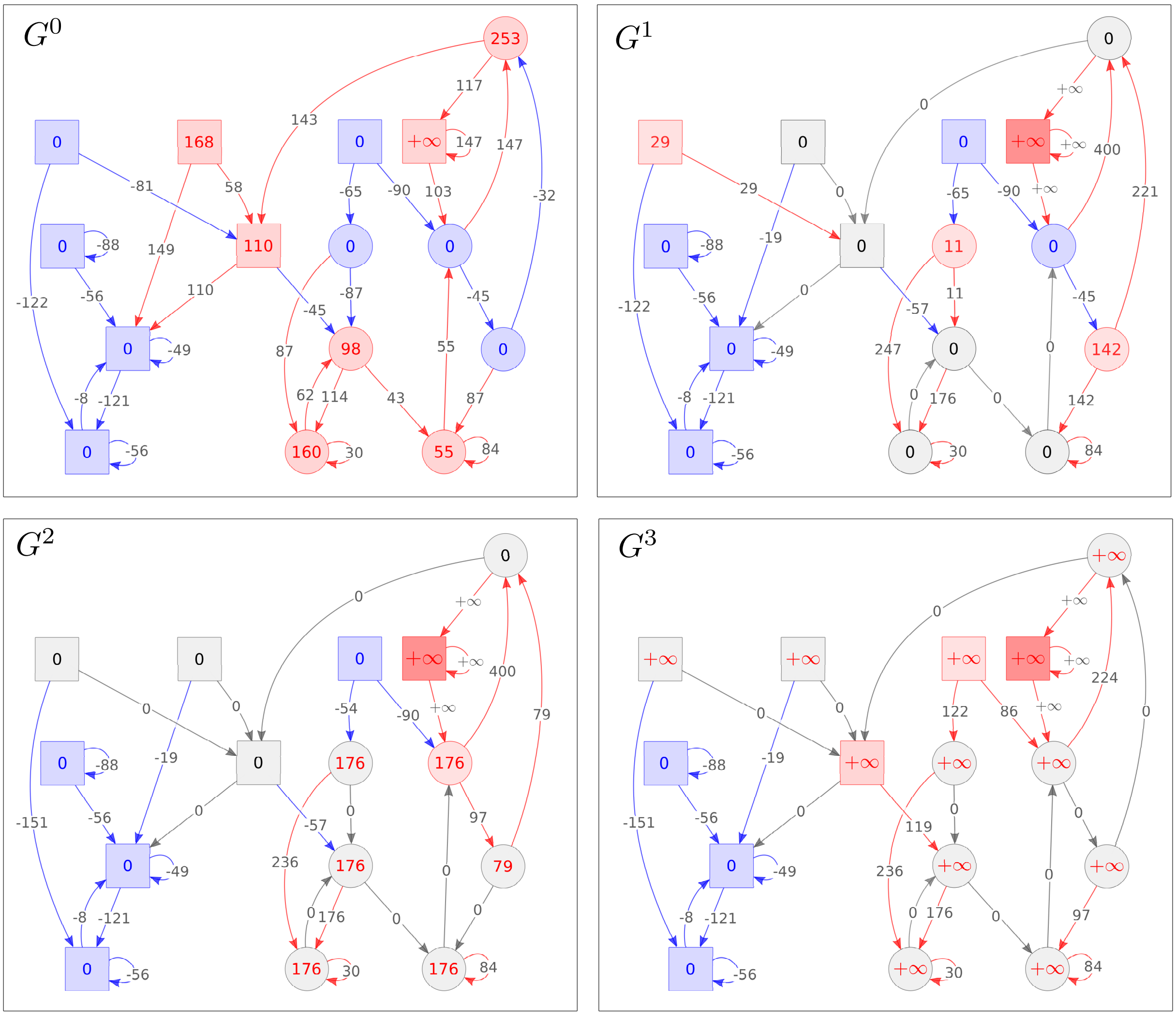


Figure 10.9: A complete execution of the ESL algorithm with four iterations over an arena of size 15 and degree 2 (which was sampled randomly). In each iteration, we indicate the SU^+-values of each vertex; vertices are coloured according to the sign of their extremal weight to improve readability. Optimal positional strategies[4] are indicated with bold arrows. Over such arenas, the average number of iterations is around 3.5; it decreases when the degree grows.

Theorem 10.5 (Extended potential reductions and energies)

Let ϕ be an extended potential such that $0 \leqslant \phi \leqslant \mathrm{Energy}^+_G$. Then we have

$$\mathrm{Energy}^+_{G^\phi} + \phi = \mathrm{Energy}^+_G.$$

Proof. Let $v \in V$. If v has infinite energy in G, then any Adam strategy τ with infinite energy in G guarantees infinite energy in G^ϕ, regardless of ϕ. If v has finite energy in G, then we pick an optimal positional strategy σ from v, and conclude by applying Theorem 10.1 in G_σ, over which ϕ is finite by assumption. □

An ESL iteration. The ESL algorithm is based on the fact that the optimal SU^+ values can be computed efficiently over arbitrary $\mathbb{Z}^\infty$ arenas, by a straightforward extension of Dijkstra's algorithm, in time $O(m + n \log n)$. This is due to the fact that only non-negative weights are considered; it is quite remarkable (and not often exploited!) that in this case, moving from graphs to games incurs no loss in complexity for Dijkstra's algorithm.

This was first observed by Khachiyan, Gurvich and Zhao[5] [KGZ06] (Theorem 1 therein, where SU^+ corresponds to case (i) with blocking systems $\mathcal{B}_2$). A similar algorithm was also given by Schewe [Sch08] with complexity $O(m_{P*} + |P^*| \log |P^*|)$, where $P^* \subseteq V$ is the set of vertices with positive SU^+ (this also corresponds to the set P^* from the GKK algorithm, see Section 3). Luttenberger [Lut08] uses a variant of the Bellman-Ford algorithm, with higher complexity $O(mn)$.

Theorem 10.6 (Efficient computation of an iteration by Dijkstra's algorithm)

The positively safe extended potential $\mathrm{SU}^+_G : V \to \mathbb{Z}^\infty$ can be computed in time $O(m+n \log n)$ over simple arenas.

We present the algorithm of [KGZ06], full details can be found in their paper (or in [Sch08]).

Proof. We start by determining in linear time $O(m)$ the set N of vertices with negative extremal weight; these have SU^+ value 0. We then initialise F, the set of vertices over which SU^+_G is known, to N. Note that all remaining Eve vertices have only non-negative outgoing edges, and all remaining Adam vertices have (at least) a non-negative outgoing edge.

We then iterate the two following steps illustrated in Figure 10.10. (A complexity analysis is given below.)

1. If there is an Adam vertex $v \notin F$ all of whose non-negative outgoing edges $v \xrightarrow{t} v'$ lead to F, set $\mathrm{SU}^+_G(v)$ to be the maximal such $t + \mathrm{SU}^+_G(v')$, add v to F, and go back to 1.

2. Otherwise, let $v \xrightarrow{t} v'$ be an edge from $V_{\text{Eve}} \backslash F$ to F (it is necessarily positive) minimising $t + \mathrm{SU}^+_G(v')$; set $\mathrm{SU}^+_G(v) = t + \mathrm{SU}^+_G(v')$, add v to F and go back to 1. If there is no such edge, terminate.

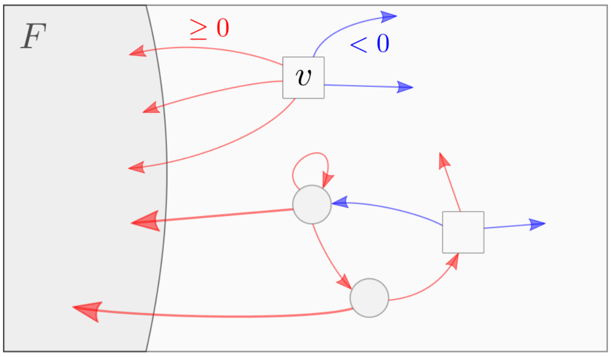


Figure 10.10: The game version of Dijkstra's algorithm; blue edges are negative and red ones are non-negative. If there is a vertex such as v (it belongs to Adam and all edges pointing out of F are < 0), one may set the value of v. Otherwise, set the value of an Eve vertex v minimising $t + \mathrm{SU}^+(v')$ over edges $v \xrightarrow{t} v'$ going to F; if there is no such edge, terminate (Adam can force seeing $\geqslant 0$ edges forever).

[5] We are grateful to Alexander Kozachinskiy for pointing out this reference to us.

After the iteration has terminated, there remains to deal with ${}^{c}F$, which is the set of vertices from which Adam can ensure to visit only non-negative edges forever. Since the arena is assumed to be simple (no simple cycle has weight zero) it holds that SU_G^+ is ∞ over F, and we are done[6].

As usual, by storing the number of edges outgoing from Adam vertices in ${}^{c}F$ to F, step 1 induces only a total linear runtime $O(m)$. For step 2, one should store, for each $v \in V_{\text{Adam}} \backslash F$, the edge towards F minimising $t + \mathrm{SU}_G^+(v')$ in a priority queue. Using a Fibonacci heap as was first suggested by Fredman and Tarjan [FT84] for Dijkstra's algorithm lowers the complexity from $O(m \log n)$ to $O(m + n \log n)$. □

The ESL algorithm iteratively applies the extended potential reduction given by SU_G^+, and terminates when an arena G_∞ is reached such that $\mathrm{SU}_{G_\infty}^+$ takes only values 0 and ∞ (stated differently, G_∞ is reduced as in Section 2). We then have $\mathrm{SU}_{G_\infty}^+ = \mathrm{Energy}_{G_\infty}^+$, and as for the GKK algorithm we recover Energy_G^+ thanks to Theorem 10.5 since extended potential reductions compose by addition.

Note that nN is not "hardcoded" in the ESL algorithm as it is in the BCDGR algorithm. Vertices with infinite energy are now detected whenever a strategy is available for Adam which avoids seeing any negative weight (in this case, the energy is necessarily infinite thanks to our simplicity assumption).

Termination of the ESL algorithm. There remains to prove termination of the ESL algorithm, for which we rely on the following lemma (which is analogous to Lemma 10.3 for the GKK algorithm). As previously, we let G' denote the arena obtained after the potential reduction, and use N and N' for the sets of vertices with extremal weight < 0.

Lemma 10.6 (Evolution of N)

We have $N \supseteq N'$.

Proof. We show that ${}^{c}N \subseteq {}^{c}N'$. Let $v \notin N$. If v has SU^+-value ∞ in G then it has only outgoing edges of weight $+\infty$ in G' therefore it cannot belong to N'; we thus assume otherwise.

- Assume v belongs to V_{Adam}. Let τ be a SU^+-optimal strategy in G, and let $e = v \xrightarrow{t} v' = \tau(\varepsilon)$. Since $v \notin N$ we have $t \geqslant 0$ and $\mathrm{SU}^+(v) = t + \mathrm{SU}^+(v')$. Hence we have $t^\phi(e) = 0 \geqslant 0$ so $v \notin N'$.

- Assume now that $v \in V_{\text{Eve}}$. We have for all $e = v \xrightarrow{t} v' \in E$ that $t \geqslant 0$ hence $\mathrm{SU}^+(v) \leqslant t + \mathrm{SU}^+(v')$, and thus $t^\phi(e) \geqslant 0$, the wanted result. □

We now let $G^0 = G$ denote the initial $\mathbb{Z}$-arena, and for each $j \geqslant 0$ we let $\phi^j = \mathrm{SU}_{G^j}^+$ and $G^{j+1} = (G^j)^{\phi^j}$ be the $\mathbb{Z}^\infty$-arena obtained after j iterations of the ESL algorithm. As before, we also let $\Phi^j = \phi^0 + \cdots + \phi^{j-1}$, and we have $G^j = (G^0)^{\Phi^j}$.

The lemma directly gives $N^0 \supseteq N^1 \supseteq \ldots$ and therefore vertices v' in N^j satisfy $\Phi^j(v') = 0$. Now if v is a vertex such that $\phi^j(v) = \mathrm{SU}_{G^j}^+$ is finite, then by definition there is a simple path in G from v to some $v' \in N^j$ with Φ^j-modified weight $\phi^j(v)$. This rewrites as

$$0 \leqslant \phi^j(v) = -\Phi^j(v) + \underbrace{\Phi^j(v')}_{0} + \underbrace{\sum_{i=0}^{k} t_i}_{\leqslant nN} ,$$

[6] If the arena is not simple, one must additionally solve a Büchi game, and the complexity of the iteration is increased. We believe that this increased cost can be amortised overall, but give no details for this claim.

and thus $\Phi^j(v) \leqslant nN$. Stated differently, finite values remain $\leqslant nN$, which guarantees termination in at most $O(n^2N)$ iterations.

Using an implementation similar to the BCDGR algorithm, where the data structures (and the set P^* of increasing vertices) is stored from an iteration to the next, one may lower the global complexity upper bound to $O((m + n \log n)nN)$, at least for simple arenas. Such an implementation was already suggested by Schewe [Sch08], but no complexity bound is given for mean-payoff games[7].

6.2 The alternating ESL algorithm

Note that the GKK potential reduction is bi-safe, whereas the BCDGR and ESL potentials are only positively safe, and in this sense asymmetric. Actually, it is not hard to see that the GKK potential is precisely the largest *constant*[8] bi-safe potential, which gives another way of presenting the GKK algorithm. It would be interesting to forge (non-constant) bi-safe potential which can be computed efficiently, which would lead to other symmetric algorithms.

We take a different direction and consider a potential update which is neither positively nor negatively safe: the one obtained by first applying the ESL update SU^+, and then its dual SU^- – defined by summing (non-positive) weights until the first positive weight is seen – and so on. This requires extending arenas and potentials to $\mathbb{Z} \cup \{-\infty, +\infty\}$, which is again a formality; we give no more details for the sake of conciseness. A complete example is given in Figure 10.11.

We believe that this algorithm is interesting for four reasons. First, it is simple to describe, implement or run by hand: simply alternate between SU^+ and SU^- potential updates – each of which is not much harder than Dijkstra's algorithm – and terminate when a potential is reached which takes values only $\pm\infty$ (over simple arenas). Second, it is completely symmetric, and only very few such algorithms are known[9]. Third, we have observed empirically (see below) that even fewer iterations are often performed (and especially for parity games), compared to the ESL algorithm. Fourth, we are not able to establish its termination with the currently available tools (even over parity games), and so far have failed to understand the subtle combinatorics it involves.

6.3 Empirical comparisons

We discuss empirical comparisons of GKK, ESL and alternating ESL (AESL) over energy games, and also compare ESL and AESL with Oink's implementations [Dij18b] of Zielonka's algorithm over parity games. Our initial motivation for these simulations was to visualise AESL over small examples, and moreover empirically establish its termination. We have not implemented the DKZ algorithm, but believe its behaviour to be similar to GKK. The BCDGR algorithm is already impractical over arenas of size $\leqslant 100$ with weights up to 1000. Our code is publicly available at

`https://github.com/PierreOhl/parity_games_solving.`

[7] Schewe's presentation is done over parity games; no complexity bound is given for mean-payoff games, which are only mentioned in a footnote. This is quite unfortunate, since his algorithm would have improved on the state of the art at that time (and also suggests the fruitful link with energy games, which was exploited by [BCD+11] only a few years later).

[8] By "constant", we mean that the potential takes only one nonzero value (or equivalently up to shifting, only two different values).

[9] As far as we are aware, Zielonka's algorithm [Zie98], the GKK algorithm, the symmetric strategy improvement of Schewe, Trivedi and Varghese [STV15] (which is completely different), and variations over these three algorithms, cover all known symmetric algorithms for parity and energy games to date.

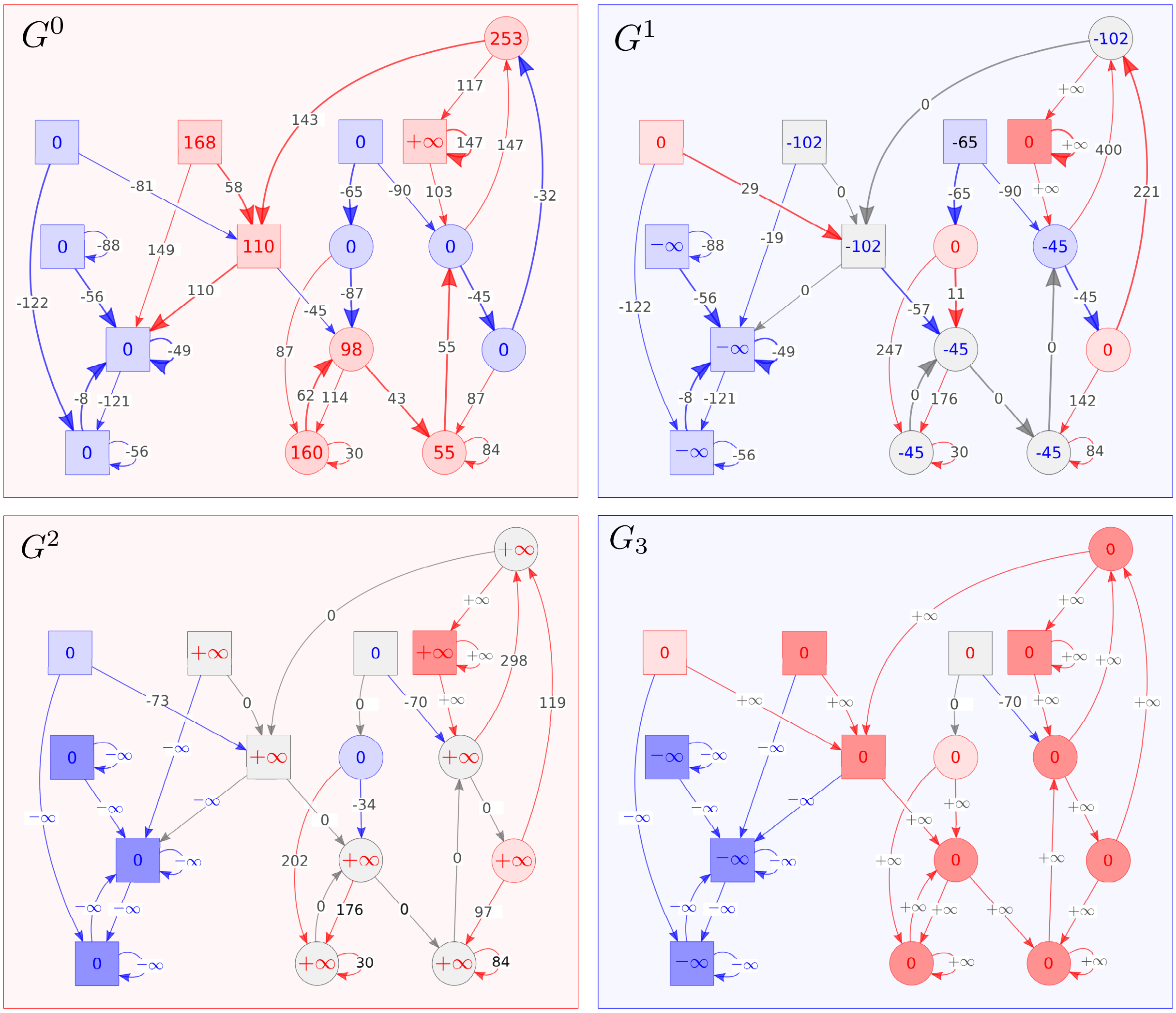


Figure 10.11: Execution of the alternating ESL algorithm on the arena of Figure 10.9, starting with a positive iteration. In arenas with an even index (red boxes), SU^{+}-values are displayed, whereas the dual SU^{-}-values are computed for odd indices. Bold arrows represent optimal positional strategies. The algorithm converges in 5 iterations, we do not depict G^4 for conciseness (the three remaining vertices converge to ∞).

Our implementations are done in Python, and all are quite naive; for instance, our implementation of the Dijkstra-like algorithm for computing SU^{+} (see Theorem 10.6) does not properly use heaps and only built-in Python data structures (int and list) are used. It is likely that several orders of magnitude could be gained in terms of runtime by using better implementations (more details are given below).

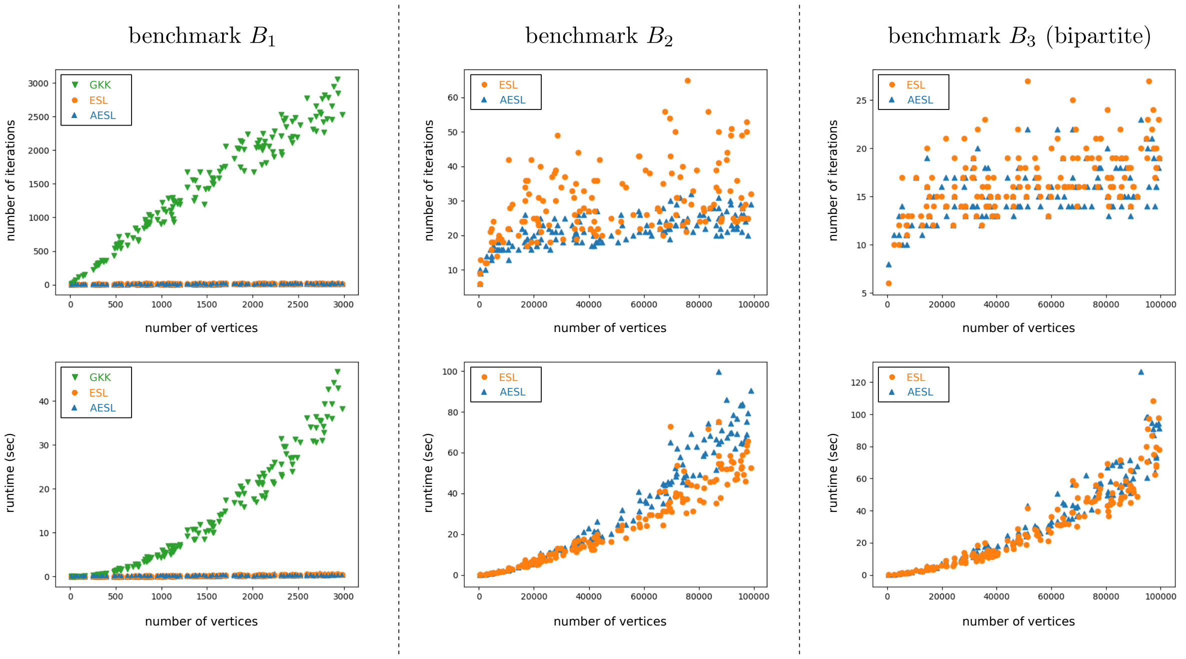
Nevertheless, we may still comment on the number of iterations, and obtained runtimes are small enough to conclude that both the ESL and alternating ESL algorithms appear to be very robust, at least over random instances. Such conclusions were also given by Schewe [Sch08] (for parity games) and Meyer and Luttenberger [ML16], who implemented an algorithm similar to ESL over GPUs.

In all experiments below, we have run the algorithms on random arenas of degree two: for each vertex, two successors are sampled uniformly at random, and colours (weights or priority) are sampled uniformly at random in a given range. Sometimes, arenas are also chosen to be bipartite (successors of V_{Eve} belong to V_{Adam} and vice-versa) to avoid having components which are trivially winning. We have observed that augmenting the degree (even degree 3, and even over bipartite games) considerably decreases the number of iterations and the runtime of all algorithms.

Comparison of algorithms for energy games. We start by comparing the different iterative algorithms for energy games. We discuss three benchmarks B_1, B_2, B_3 of 150 arenas each. (We lack of more structured benchmarks; we stress again that these are preliminary empirical results.)

- In B_1, arenas have size from 1 to 3000, degree 2, and weights are drawn in the range $[-1000, 1000]$.
- In B_2 and B_3, arenas have size from 1 to 100000, degree 2, and weights are drawn in the range $[-10^{10}, 10^{10}]$. Benchmark B_3 is moreover comprised only of bipartite arenas.

Results are depicted in Figure 10.12.


benchmark B_1

benchmark B_2

benchmark B_3 (bipartite)

Figure 10.12: Comparison of the energy games algorithms GKK, ESL and AELS on benchmarks B_1, B_2 and B_3. The number of iterations of GKK grows linearly with n, whereas it is almost constant for both ESL and AESL, which scale up to much larger arenas. It appears that AESL is a bit more robust (in number of iterations) than ESL over generic arenas, however this difference is not observed over bipartite arenas (we cannot explain this fact). The runtimes displayed by ESL and AESL are similar.

We observe that the ESL and AESL outperform GKK, which does not scale to benchmarks B_2 and B_3. Both ESL and AESL make only a very small number (almost constant) of iterations, even over large arenas. With degree 5, this number even drops to typically $\leqslant 10$ iterations for both algorithms (on arenas of size up to 100000 as in B_2 and B_3). Similar results are presented in [Sch08] (for parity games with small numbers of colours), and the implementation of [ML16] scales to arenas with 40 million vertices (but the number of iterations is not discussed).

Comparison with Oink's implementations of Zielonka's algorithm. We have also compared ESL and AESL with Oink's implementations [Dij18b] of the Zielonka algorithm over parity games.

To run algorithms implemented in Oink, we use arenas where priorities label the vertices (which are easily transferred to our setting). We use three benchmarks B_4, B_5, B_6 of 100 arenas each generated uniformly at random as above. For all parity arenas we generate, we draw priorities at random between 1 and n (there are linearly many priorities). The three benchmark are comprised only of bipartite arenas of degree 2.

- In B_4, arenas have size from 1 to 50000.
- In B_5, all arenas have size 20000.
- In B_6, all arenas have size 80000.

Results are depicted in Figure 10.13. Again, our implementation here is very naive: we work directly with weights of the form $(-n)^p$ in Python's native encoding of integers. For B_6, this corresponds to integers of magnitude up to $80000^{80000} \simeq 10^{400000}$. We ran two implementations of Zielonka's algorithm in Oink, namely UZLK and ZLK (sequentially). The first is more basic whereas the second one includes several optimisations; details can be found in [Dij18b]. Results are displayed in Figure 10.13.

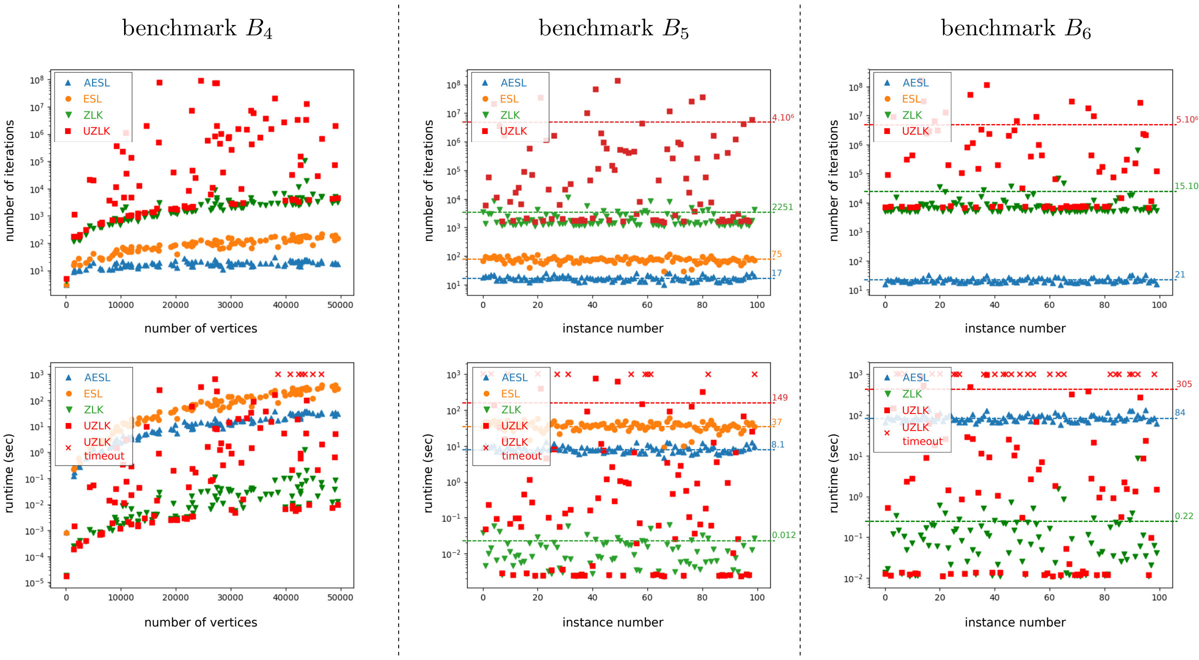


Figure 10.13: Comparison of ESL, AELS and Oink's UZLK and ZLK. In benchmark B_5, UZLK times out (1000 sec) for 12 instances. In benchmark B_6, ESL (not displayed) times out over all instances, and UZLK times out over 26 instances. For B_5 and B_6, average runtimes (which include timeouts) and number of iterations (which do not) are displayed.

We observe that over random parity games, AESL performs much better than ESL, and the number of iterations performed remains low even for large instances. Both ESL and AESL appear to be more robust than UZLK, which frequently displays high runtimes. Although they are consistently outperformed, we also observe that going from B_5 to B_6, the average runtime for AESL is roughly multiplied by 10, whereas it is multiplied by 20 for ZLK, which is an indication that AESL might scale to larger parity games.

There are many ways in which our implementation could be improved, each of which might save several orders of magnitude and bridge the runtime gap with ZLK. The four most notable are the following.

- Port it to C++ (or another lower level language), and use appropriate data structures rather than Python's lists and integers to manipulate data.
- Improve the implementation of Dijkstra's algorithm by properly using heaps (or Fibonacci heaps).
- Save time by using information (valid edges/vertices) from an iteration to the next (this is currently done for ESL but not AESL).
- In the case of parity games, use adapted structures to manipulate integers with few nonzero bits rather than words of size $O(\log(n^n)) = O(n \log n)$ manipulated by Python's native integers.

We believe that the very low number of iterations displayed over random games is a good motivation for developing better implementations. It is also encouraging that our naive implementations can already compete with UZLK. We would also be eager to run the algorithms (especially AESL) over structured benchmarks, for which a higher number of iterations should be expected; this is left to future work.

7 Conclusion and perspectives

In this chapter, we have further investigated algorithms for solving mean-payoff games. We have argued that the two natural monotonic graphs correspond to potential reductions, and proposed to look at these in a symmetric fashion.

Symmetric study of GKK algorithm. First, we have shown that the GKK algorithm of Gurvich, Karzanov and Khachiyan admits a completely symmetric presentation over simple arenas, which are those admitting only nonzero simple cycles. This allowed us to present a novel symmetric analysis, revealing a bound of $N + E^+ + E^- + 1$ on the number of iterations (for simple arenas). This new bound improves on the state-of-the-art runtime bound $O(nmN)$ by taking into account the structure of the game.

We have also established a state-of-the-art combinatorial bound $2^{n/2}$ on the number of iterations of the GKK algorithm, by adapting the technique of Dorfman, Kaplan and Zwick [DKZ19]. Similar examples for which the $2^{n/2}$ bound is matched are given in the full version of [DKZ19] for the DKZ algorithm, and can also be found[10] in [BV01] for the GKK algorithm.

It would be interesting to further study the GKK algorithm, and we propose three directions for future work.

1. Could a similar symmetric analysis be extended to non-simple arenas? Could one at least obtain the $O(mnN)$ bound for the GKK algorithm (potentially by reusing information from one iteration to the next, as in value iterations) as is suggested by the results of [DKZ19]?
2. Recall that the state-of-the-art pseudopolynomial algorithm for the mean-payoff value or strategy synthesis problems is due to Comin and Rizzi [CR16] and runs in time $O(mn^2N)$. Very roughly, it is based on repeatedly solving energy games with higher and higher energy levels using the BCDGR algorithm. Could one instead use the GKK algorithm as a subroutine, to improve the complexity (at least for simple arenas)? Could a symmetric algorithm be designed in this manner?

[10] It appears from [GKK88] that such examples are due to Lebedev.

3. Could we use the GKK algorithm (potentially, in combination with Zielonka's) to derive new (attractor-based) algorithms for mean-payoff parity games?

ESL and AESL algorithms. Second, we have used the same tools to present a variation on an algorithm of Schewe [Sch08], which we called the ESL algorithm. Roughly speaking, it accelerates the BCDGR value iteration by running, in each iteration, a game-theoretic version of Dijkstra's algorithm working in almost linear time $O(m + n \log n)$. We believe that there is value in understanding this algorithm better, since it is without doubt the most efficient one in practice for solving mean-payoff games. Schewe's algorithm is moreover an important piece of the successful LTL-synthesis tool STRIX's [MSL18; LMS20].

We have also proposed a natural symmetric variant, which alternates the potential reduction computed by Dijkstra's algorithm and its dual. We have observed empirically that it performs very few iterations over random arenas induced by parity games. However, we have not been able to derive its termination from our current toolset. Beyond more serious implementations and benchmarking, the directions we envisage for our future work on the ESL and AESL algorithms are the following.

1. Obviously, we want to understand the termination of the AESL algorithm.

2. We want to investigate combinatorial bounds for the ESL algorithm (which could also help for the first item).

3. We want to understand how the ESL algorithm behaves on Friedmann's examples [Fri09] (most likely, in an exponential fashion, just as Schewe's version). In particular, can we describe their dynamics in terms of potential reductions?

4. It would also be very interesting to further study applicability of these two algorithms. How could they be optimised, or parallelised? Could their behaviour be understood specifically on parity games? Could synthesis frameworks (such as STRIX) benefit from our simpler formalism? Could they be used on extensions of mean-payoff games?

5. Last, could we identify a wider (abstract) class of monotonic graphs over which the value iteration algorithm can be sped up by a similar game-extension of Dijkstra's algorithm?

We also hope that our work in this chapter participates in popularising the GKK and ESL algorithms both of which appear to us as very natural and flexible methods for solving energy (or parity) games.

11 Attractor-based algorithms and parity bi-progress measures

We now focus on attractor-based algorithms for parity games; we refer to the general introduction for more context. Although we also like to present the GKK algorithm as attractor-based, this chapter is completely independent from the previous one. Informally, the reason is that the nature of the symmetry considerably differs between parity and mean-payoff games: while in the latter, positive weights compete with negative weights, the symmetry is more *interleaved* in the former, where even priorities compete with odd ones.

In the previous chapter, we exploited the symmetry of mean-payoff games by looking at a single progress measure (or potential) from the point of view of both players. Here, it appears to be more natural[1] to consider *interleaving* two given progress measures, one from the point of view of each player. We will call the resulting object a *(parity) bi-progress measure*.

The observation which motivated this chapter is that Zielonka's positionality proof [Zie98] directly relies on ordinals: an (asymmetric) construction of a family of parity-universal monotonic graphs (which is different from the one of Emerson and Jutla [EJ91] or equivalently Walukiewicz's signatures [Wal96], used in Chapter 5) can be extracted from his work. This construction can be related to Klarlund's *quasi progress measures* [Kla91] (which he introduced for complementing Streett automata), also instrumental (under the name *lazy progress measures*) in the recent work of Daviaud, Jurdziński and Lehtinen [DJL19].

More recently, Parys [Par19] gave a quasipolynomial version of Zielonka's algorithm, improved by Lehtinen, Schewe and Wojtczak [LSW19] by using the universal tree of Jurdziński and Lazić [JL17] (see also their joint full version [LPS+21]), and adapted to be parameterisable by arbitrary universal trees by Jurdziński and Morvan [JM20]. This reveals a combinatorial relationship between (fixpoint-based, asymmetric) value iterations and (symmetric) attractor-based algorithms: both rely on universal trees. We propose to further explore the link between these two paradigms.

Contributions and outline. The first insight we exploit is that value iterations naturally compute attractors: for instance, if $d+1$ is the largest odd priority appearing in the arena, then after i iterations any vertex at depth $\leqslant i$ in the Adam attractor to $d+1$ (this is formally defined below) has its $d+1$ coordinate $\geqslant 1$ in the progress measure. Building up on this observation, we prove a *simulation result*, which roughly establishes that the attractor-based algorithm of [JM20] can be obtained by running synchronously running a value iteration algorithm for each player, with virtually no interaction between the two, except that the iteration stops whenever each vertex is mapped to

[1] As we will see, this is not only natural but also completely generic: one can interleave parity progress measures regardless of the underlying monotonic graph.

either of the two $\top$ elements. Perhaps surprisingly, to simulate the algorithm of [JM20] parameterised with trees T^{odd} and T^{even}, one simply uses the construction $\mathcal{L}_{T^{\text{odd}}}$ from Chapter 5, based on signatures, and its dual.

Although it is intuitively simple, establishing the simulation result requires tedious formalities. In particular, it is adequate to rely on a streamlined, non-recursive presentation of the algorithm of [JM20] which we call *attractor-labelling*: the algorithm is defined iteratively as a depth-first traversal of the interleaving T (defined below) of T^{odd} and T^{even}. The first section is devoted to such a presentation, and the second one establishes the simulation result (the main result of [JM20], namely correctness of their algorithm, is obtained as a consequence).

The reader might be surprised at this point: two contradictions immediately arise. First, the algorithm of [JM20] has complexity (up to polynomial factors) roughly $|T^{\text{odd}}||T^{\text{even}}|$, while the synchronous value iteration has complexity roughly $|T^{\text{odd}}| + |T^{\text{even}}|$ (or even $\min(|T^{\text{odd}}|, |T^{\text{even}}|)$). This advantage comes from *monotonicity* of the value iteration approach: recall from the general introduction (Figure 2 therein) that information (winning strategies in subgames) is repeatedly discarded in attractor-based algorithms, whereas this is not the case in value iterations. Stated differently, our simulation result explains that progress measures can be seen as adequate data structures for running the algorithm of [JM20] more efficiently, by avoiding to discard information.

Second, Zielonka's algorithm is well-known to frequently perform subexponentially many (even polynomially many) iterations, while this is not the case of value iterations (even when ran in parallel as above). This is not contradictory because, as explained in [JM20], even when T^{odd} and T^{even} are chosen to be complete n-ary trees of height $d/2$, their algorithm does not match Zielonka's and perform more steps. (Parys' algorithm cannot be matched either, even with adequate tree parameters.)

Section 3 explores the generic structure of parity bi-progress measures, and formalises a general way of using information (validity) from one progress measure to accelerate the other. Finally, in Section 4, we give a detailed sketch proving that by using Zielonka's construction of monotonic graphs (akin to [Kla91] or [DJL19] as explained above), combined with our acceleration, one may simulate Zielonka's algorithm, or even derive progress measure-based non-discarding variants of it. This opens different perspectives for further work on this front, which are discussed in Section 5.

This chapter is the fruit of a joint work with Marcin Jurdziński, Rémi Morvan and K. S. Thejaswini, a preprint is available at [JMO+20]. The presentation we make here is very different and the precise results we derive are incomparable, but the underlying ideas are the same. We are also grateful to Antonio Casares, Thomas Colcombet, Nathanaël Fijalkow and Olivier Serre for many stimulating discussions.

1 Universal attractor-based algorithm of [JM20]

In this section, we present the algorithm of [JM20] in a streamlined fashion, which is formally heavier but more adequate to our needs. Its (non-trivial) correctness will be derived in the next section, as a consequence of our simulation result. We start by formally introducing attractors.

Reachability games and attractors. Fix a finite $\{\text{good}, \text{wait}\}$-arena G and consider the reachability game: Eve wants to visit a good-edge. Recall from Chapter 2 that the winning region is given by a fixpoint of the form $\psi : V \to \omega$, which can be explicited setting $A_{\leqslant i} = \psi^{-1}([0,i])$ by

$$A_{\leqslant i} = \left\{ v \in V_{\text{Eve}} \mid \exists\, v \xrightarrow{c} v' \text{ with } \begin{array}{c} c = \text{good or} \\ v' \text{ in } A_{\leqslant i-1} \end{array} \right\} \cup \left\{ v \in V_{\text{Adam}} \mid v \xrightarrow{c} v' \text{ implies } \begin{array}{c} c = \text{good or} \\ v' \text{ in } A_{\leqslant i-1} \end{array} . \right\}$$

for all i. See Figure 11.1 for an illustration.

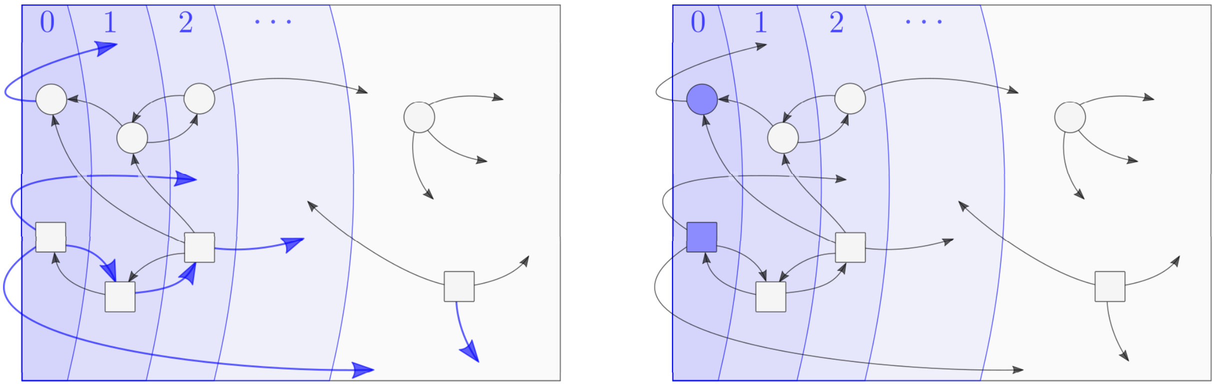


Figure 11.1: On the left, the Eve attractor to bold blue edges. On the right, the Eve attractor to blue vertices.

Since $|V| = n$ there is $r \leqslant n$ such that $A_{\leqslant r} = A_{\leqslant r-1}$, which, as we have already proved (Lemma 2.4), implies $A_{\leqslant r+i-1} = A_{\leqslant r-1}$ for all i, and therefore $A_i = A_{\leqslant i} \backslash A_{\leqslant i-1}$ is empty if $i \geqslant r$. It is not hard to see that the global value iteration algorithm (from the point of view of Adam) for solving the safety game in the safety-universal monotonic graph of size one (see Chapter 2) iteratively computes A_i, for $i < r$, with linear runtime $O(m)$.

Given a C-arena G and a subset E' of its edges, we call *Eve-attractor to E' in G* the winning region in the reachability game obtained from G by colouring edges in E' with good and others with wait. It is the disjoint union of the non-empty sets $A_0, A_1, \ldots, A_{r-1}$ described above, which we call the *Eve-attractor layers to E' in G*, and we say that their number r is the *attractor-depth*.

We let $\mathrm{Attr}^{\mathrm{Eve}}_G(E')$ denote the Eve attractor to E' in G. Observe that by definition of safety games, the pregraph obtained by removing edges from E' and restricting to the complement of the attractor is has no sink in general.

Given a subset $V' \subseteq V$ of the vertices, we also call *Eve-attractor to V' in G* the attractor $\mathrm{attr}^{\mathrm{Eve}}_G(E')$ where $E' = \bigcup_{v' \in V'} \mathrm{Out}(v')$ is the set of edges outgoing from vertices in V'. For convenience, we use the same notation $\mathrm{Attr}^{\mathrm{Eve}}_G(V')$ and also use $A_0, \ldots, A_{r-1}$ to denote the attractor layers. Note that we have in this case that $A_0 = V'$ and that $A_{\leqslant i+1}$ is comprised of Eve vertices with an outgoing edge towards $A_{\leqslant i}$ and of Adam vertices all of whose outgoing edge are towards $A_{\leqslant i}$. In the case of vertex attractors, we define the depth to be $r-1$, for instance if the attractor is V' itself then the depth is 0.

Adam-attractor (of subsets of edges or vertices) are defined dually.

1.1 A non-recursive presentation

For the sake of discussion and clarity, we include a copy of Jurdziński and Morvan's pseudo-code, which is based on two mutually recursive procedures, as is the case of the Zielonka algorithm.

It is more convenient for our needs to make completely explicit the structure of the recursive calls of the algorithm. This requires the notion of (tree) interleaving, which was already discussed in [JM20]. We fix d to be an even integer.

Interleaving. Recall that we denote $u = (u_{d-1}, u_{d-3}, \ldots, u_1)$ when interpreting tuples in $\omega^{d/2}$ as occurrences of odd priorities. Given a tuple $u = (u_d, u_{d-1}, \ldots, u_1) \in \omega^d$, we let $u^{\mathrm{odd}} = (u_{d-1}, \ldots, u_1) \in \omega^{d/2}$ denote its projection on odd coordinates, and likewise we let $u^{\mathrm{even}} = (u_d, u_{d-2}, \ldots, u_2)$ denote its projection on even coordinates.

```
procedure $\mathtt{Univ}_{\mathrm{Even}}(\mathcal{G}, d, \mathcal{T}^{\mathrm{Even}}, \mathcal{T}^{\mathrm{Odd}})$:
    let $\mathcal{T}^{\mathrm{Odd}} = \langle \mathcal{T}_1^{\mathrm{Odd}}, \mathcal{T}_2^{\mathrm{Odd}}, \dots, \mathcal{T}_k^{\mathrm{Odd}} \rangle$
    $\mathcal{G}_1 \leftarrow \mathcal{G}$
    for $i \leftarrow 1$ to $k$ do
        $D_i \leftarrow \pi^{-1}(d) \cap \mathcal{G}_i$
        $\mathcal{G}_i' \leftarrow \mathcal{G}_i \setminus \mathrm{Attr}^{\mathcal{G}_i}_{\mathrm{Even}}(D_i)$
        $U_i \leftarrow \mathtt{Univ}_{\mathrm{Odd}}\left(\mathcal{G}_i', d-1, \mathcal{T}^{\mathrm{Even}}, \mathcal{T}_i^{\mathrm{Odd}}\right)$
        $\mathcal{G}_{i+1} \leftarrow \mathcal{G}_i \setminus \mathrm{Attr}^{\mathcal{G}_i}_{\mathrm{Odd}}(U_i)$
    return $V^{\mathcal{G}_{k+1}}$
```

```
procedure $\mathtt{Univ}_{\mathrm{Odd}}(\mathcal{G}, d, \mathcal{T}^{\mathrm{Even}}, \mathcal{T}^{\mathrm{Odd}})$:
    let $\mathcal{T}^{\mathrm{Even}} = \langle \mathcal{T}_1^{\mathrm{Even}}, \mathcal{T}_2^{\mathrm{Even}}, \dots, \mathcal{T}_\ell^{\mathrm{Even}} \rangle$
    $\mathcal{G}_1 \leftarrow \mathcal{G}$
    for $i \leftarrow 1$ to $\ell$ do
        $D_i \leftarrow \pi^{-1}(d) \cap \mathcal{G}_i$
        $\mathcal{G}_i' \leftarrow \mathcal{G}_i \setminus \mathrm{Attr}^{\mathcal{G}_i}_{\mathrm{Odd}}(D_i)$
        $U_i \leftarrow \mathtt{Univ}_{\mathrm{Even}}\left(\mathcal{G}_i', d-1, \mathcal{T}_i^{\mathrm{Even}}, \mathcal{T}^{\mathrm{Odd}}\right)$
        $\mathcal{G}_{i+1} \leftarrow \mathcal{G}_i \setminus \mathrm{Attr}^{\mathcal{G}_i}_{\mathrm{Even}}(U_i)$
    return $V^{\mathcal{G}_{\ell+1}}$
```

Figure 11.2: The pseudo-code from [JM20]. The notations are not formally adapted, and the formalism is slightly different, mainly because games are vertex-coloured.

Conversely, given $u^{\mathrm{even}} = (u_d, \dots, u_2), u^{\mathrm{odd}} = (u_{d-1}, \dots, u_1) \in \omega^{d/2}$, we let $u = \iota(u^{\mathrm{even}}, u^{\mathrm{odd}}) \in \omega^d$ be given by $u = (u_d, u_{d-1}, \dots, u_2, u_1)$. We always use the notation u for the interleaving of u^{even} and u^{odd}, and conversely u^{even} and u^{odd} for the projections of u.

Now given two trees of height $d/2$, $T^{\mathrm{even}} \subseteq \omega^{d/2}$ and $T^{\mathrm{odd}} \subseteq \omega^{d/2}$, respectively interpreted as describing occurrences respectively of even and odd priorities, we let $T = \iota(T^{\mathrm{even}}, T^{\mathrm{odd}})$ denote their interleaving, formally

$$u \in T \qquad \Longleftrightarrow \qquad u^{\mathrm{even}} \in T^{\mathrm{even}} \text{ and } u^{\mathrm{odd}} \in T^{\mathrm{odd}}.$$

Note that although ι defines a bijection between ω^d and $(\omega^{d/2})^2$, trees obtained as interleavings have a special shape; for instance if $d = 2$, interleavings correspond to rectangles in ω^2.

Depth-first traversal. We now fix an arbitrary tree $T \subseteq \omega^d$ of height d. (T will later be taken as an interleaving, but this does not impact the definitions below.) We let $\overline{T} \subseteq \omega^{\leqslant d}$ denote the set of prefixes of T. In particular we have $T \subseteq \overline{T}$. We say that u' is *below* u if u is a prefix of u'. We call elements of $\overline{T}$ the *nodes* of T, and elements of T the *leaves* of T (which are special nodes).

The interleaving T of the two trees gives the global structure of the recursive calls in the algorithm. Consider the procedure on the left hand-side of Figure 11.2. Informally $\overline{T}$ is visited in a depth-first fashion; when reaching a node u, an attractor is computed, then a recursive call is made "below u", and then another attractor is computed, after which there are two cases: either u has a right sibling (case where $i < k$) in which case the algorithm continues from the right sibling of u, or u has no right sibling (case where $i = k$) in which case the algorithm continues from the parent of u.

Schematically this can be described as

$$\text{1st attractor at } u \to \text{ computation below } u \to \text{ 2nd attractor at } u \to \text{ computation after } u,$$

in particular, two different attractors are computed at u, at different moments in the algorithm.

We let $\overline{T}^{\uparrow\downarrow} = \{u{\uparrow} \mid u \in \overline{T}\} \cup \{u{\downarrow} \mid u \in \overline{T}\}$. In other words $\overline{T}^{\uparrow\downarrow}$ is comprised of two copies of each node of T; see Figure 11.3. We use the word *position* to refer to elements of $\overline{T}^{\uparrow\downarrow}$, which we denote with x or y. Each node $u \in \overline{T}$ is associated with two positions $u{\uparrow}$ and $u{\downarrow}$, which are associated to the two attractors computed at node u. Positions of the form $u{\downarrow}$ are called *descending* and those of the form $u{\uparrow}$ are called *ascending*; we refer to this as the *orientation* of the position.

The depth-first traversal of $\overline{T}$ is given as a linear order $\preccurlyeq$ over $\overline{T}^{\uparrow\downarrow}$. It is uniquely described by

$$u{\downarrow} \preccurlyeq u'{\downarrow} \preccurlyeq u'{\uparrow} \preccurlyeq u{\uparrow}$$

whenever u' is below u, and

$$u{\downarrow} \leqslant u'{\downarrow} \quad \Longleftrightarrow \quad u \leqslant u',$$

where $\leqslant$ denotes the lexicographical order.

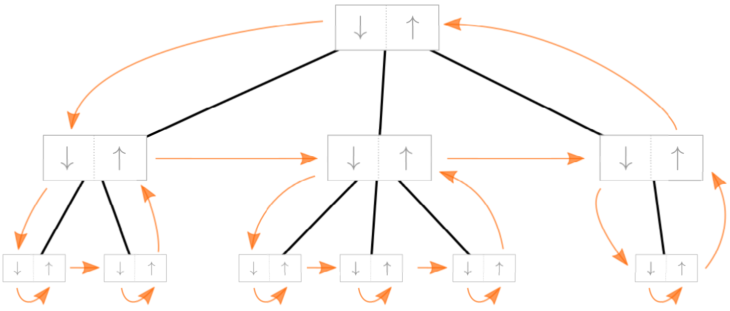


Figure 11.3: An illustration of the depth-first traversal of the tree $T = \{00, 01, 10, 11, 12, 22\}$, of height two. The tree is depicted in black, the positions are represented as cells, and the traversal, which is a linear order over positions, is given by the orange arrows.

The first element in the traversal is $\varepsilon{\downarrow}$ and the last is $\varepsilon{\uparrow}$.

We let $\mathrm{Succ}(x)$ and $\mathrm{Pred}(x)$ respectively denote the immediate successor and predecessor of x when it is defined. We employ standard terminology with respect to nodes in ordered trees:

- the *height* of a node $u = (u_d, u_{d-1}, \dots, u_p)$ is p, in particular leaves have height 1 and the root ε has height $d + 1$;
- u is the *parent* of u' if u' is below u and their heights differ by 1, and in this case u' is a child of u;
- *siblings* are nodes with a common parent;
- a sibling u' of u is a *right sibling* if it is strictly greater, and a *left sibling* if it is strictly smaller;
- the direct right sibling of u is its least right sibling if it has one; likewise the first child of u is its least child if it has one.

This allows us to explicit $\mathrm{Succ}(x)$ (except for $x = \varepsilon{\uparrow}$ which is the last position) as follows

$$\mathrm{Succ}(u{\downarrow}) = \begin{cases} u{\uparrow} & \text{if } u \text{ is a leaf} \\ u'{\downarrow} & \text{where } u' \text{ is the first child of } u\text{, otherwise} \end{cases}$$

$$\mathrm{Succ}(u{\uparrow}) = \begin{cases} u'{\downarrow} & \text{where } u' \text{ is the direct right sibling of } u \text{ if it has one} \\ u'{\uparrow} & \text{where } u' \text{ is the parent of } u\text{, otherwise.} \end{cases}$$

Similar formulas can be obtained for $\mathrm{Pred}(x)$ by inverting the above ones.

Attractor-labelling. We now additionaly fix a $[1, d]$-arena G. Given a subset of vertices $X \subseteq V$ and an priority p we let G^p_X denote the restriction of G to vertices in X and edges of priority $\leqslant p$, and we let E^p_X denote the set of edges of the form $X \xrightarrow{p} X$ in G.

We use $\mathrm{P} \in \{\mathrm{Eve}, \mathrm{Adam}\}$ to denote either player, and $\overline{\mathrm{P}}$ for the opponent of P. Given $u \in \overline{T}$, the *player associated to* u corresponds to the priority of its height p: it is $\mathrm{P} = \mathrm{Eve}$ if p is even and

P = Adam otherwise. Note that the player associated to the root ε, as well as the player associated to leaves, is Adam, since these have odd heights.

We say that a collection of subsets of vertices $X_u, Y_u, A_x \subseteq V$, where u ranges over the set of nodes $\overline{T}$ of T and x ranges over the set $\overline{T}^{\uparrow\downarrow}$ of positions in T, is an *attractor-labelling* (of T by G) if (see illustration in Figure 11.4)

- $X_\varepsilon = V$;
- for all[2] nodes $u \in \overline{T}$ we have
  - $X_u = X_{u'} \setminus A_{x'}$ and
  - $Y_u = X_{u''} \setminus A_{x''}$,

  where $x' = \mathrm{Pred}(u\downarrow)$, $x'' = \mathrm{Pred}(u\uparrow)$, and u' and u'' are their associated nodes; and
- for all $u \in \overline{T}$, defining p to be the height of u and P to be the player associated to u, we have
  - $A_{u\downarrow} = \mathrm{Attr}^{\mathrm{P}}_{G^{\leqslant p}_{X_u}}(E^p_{X_u})$ and
  - $A_{u\uparrow} = \mathrm{Attr}^{\overline{\mathrm{P}}}_{G^{\leqslant p}_{X_u}}(Y_u)$.

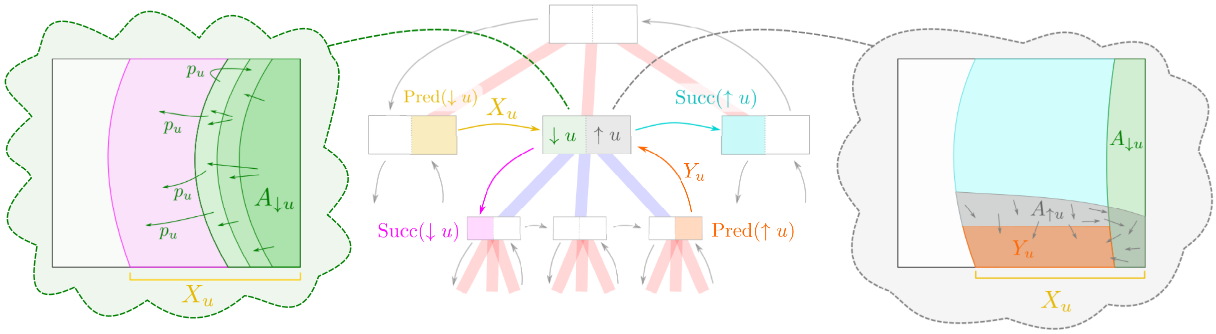


Figure 11.4: A complete depiction of the situation at a node $u \in \overline{T}$ in an attractor labelling of an arena by a tree of height d. Each node should be seen as inducing two computation cells corresponding to the two positions $u\downarrow$ and $u\uparrow$ which occur at different moments in the depth-first traversal. Each position is responsible for computing an attractor, and then passing on the information to its successor. In the figure, u is not a leaf (it has children), and has sibling on both sides. The two illustrations on the left and the right depict the computation in each cell; the neighbourhood of u in $\overline{T}$ is represented in the center.

Note that since there are no $(d+1)$-edges, we have $A_{\varepsilon\downarrow} = \varnothing$; stated differently the first step is (artificially) trivial. This has no impact on the rest of the presentation.

We will see as an easy property of attractor-labellings that for all nodes u we have $Y_u \subseteq X_u$ (see Lemma 11.2), which is why the definition of $A_{u\uparrow}$ makes sense. We say that $G^p_{X_u}$ is the *arena associated with* u.

We will often drop subscripts for clarity, when the node and/or the position is clear from context. More precisely, we will often refer to a position x and simply use $u, A, X, Y, \mathrm{P}, p$ respectively for the node associated to x, A_x, X_u, Y_u, the player associated to u and the height of u. Likewise, when we define a position x', we use $u', A', X', Y', \mathrm{P}', p'$ for the corresponding objects, avoiding the hassle

[2] The first equation does not make sense for $u = \varepsilon$, but the second one is required to hold.

of formally stating these definitions. The equational form we chose for convenience implies that we should prove the statement below.

Lemma 11.1 (Attractor-labellings are well defined)

There exists a unique attractor-labelling of T with G.

The formal proof is a bit cumbersome but only syntactical.

Proof. We show by induction on the traversal of T that

for all positions x associated with node u, there exists a unique collection $X_w, Y_{w'}, A_y$ of subsets, where y ranges over positions $\prec x$, w over nodes associated to such positions which are descending, and w' over nodes associated to those which are ascending.

We let $x \in \overline{T}^{\uparrow\downarrow}$ be a position associated with node u and assume that the above holds (which is vacuous for $x = \varepsilon\downarrow$, the first position). There are two cases according to the orientation of x.

- If $x = u\downarrow$. Then if $x = \varepsilon\downarrow$ we have $X_u = V$ and otherwise we have $X_u = X_{u'} \backslash A_{x'}$, where $x' = \text{Pred}(x)$ and u' is the associated node, which defines uniquely X_u since $u\downarrow \prec x$. Moreover A_x is defined uniquely using the equation above. This satisfies the wanted inductive hypothesis for $\text{Succ}(x)$.

- If $x = u\uparrow$. Then Y_u is defined uniquely with respect to $X_{u''}$ and $A_{x''}$, where $x'' = \text{Pred}(x)$ and u'' is the associated node. Since $x'' \prec x$, $A_{x''}$ is defined uniquely by induction, and so is $X_{u''}$ since $u''\downarrow \prec x$. Finally, A_x is defined uniquely with respect to X_u (which is defined by induction since $u\downarrow \prec x$) and Y_u.

This concludes the induction and the proof. □

Therefore we say "the" attractor-labelling of G by T.

Attractor-based iteration. The proof above suggests a procedure for computing the attractor-labelling step-by-step, following the depth-first traversal of T, and we call this procedure the *attractor-based iteration* (for G in T). We say that the attractor $A_{\varepsilon\uparrow}$ computed at the last position is its *output*, which we denote by $W \subseteq V$.

We claim that this procedure (and its output) coincides with that of [JM20], when T is set to be the interleaving of T^{even} and T^{odd} but keep this claim informal since the basic definitions differ. We state their main result.

Theorem 11.1 (Correctness of the attractor-based iteration)

If T^{even} and T^{odd} are universal then W is the winning region for Eve in G.

We give a few general properties which will be helpful later on.

1.2 First properties

The result below requires no game-theoretic argument, but only uses the fact that attractors are subsets of vertices. Note that the second item implies $Y_u \subseteq X_u$ for all nodes u, since by definition $Y_u \subseteq X_{u'}$, where u' is the greatest child of u.

Lemma 11.2 (Set-properties of attractor-labellings)

- If u' is a right sibling of u then $X_{u'} \subseteq X_u$.
- If u' is below u then $X_{u'} \subseteq X_u$.
- X_u is the disjoint union of $A_{u\downarrow}$, of $A_{u'\uparrow}$ where u' ranges over the children of u, and of Y_u.

Proof.

- Let $u' = uj$ and $u = ui$ for some $j \geqslant i$. We prove the first item by induction on $j - i$. If $j - i = 0$ then $u = u'$ and the result is clear. Otherwise, we have $X_{u'} = X_{\mathrm{Pred}(u'\downarrow)} \backslash A_{\mathrm{Pred}(u'\downarrow)}$ and the result follows by induction since the direct left sibling $u'\downarrow$ of u is a right sibling of u with a smaller difference in their last coordinate.

- We prove the second item by induction on the difference in the heights of u and u'. If their heights are the same then the nodes are equal and the result is clear. If their heights differ by one, then the leftmost child of u is a left-sibling of u', hence by the first item, $X_{u'} \subseteq X_{u''}$, which concludes since by definition $X_{u''} = X_u \backslash A_{u\downarrow} \subseteq X_u$. Otherwise, the parent u'' of u' is below u and higher than u', hence by induction $X_{u''} \subseteq X_u$, and since u'' and u' differ in height by one we conclude that $X_{u'} \subseteq X_{u''} \subseteq X_u$.

- Let $u_1, \ldots, u_k$ denote the children of u in this order (meaning, u_{i+1} is the direct right sibling of u_i). We prove by induction on i that $X_u \backslash X_{u_i}$ is the disjoint union of $A_{u\downarrow}$ and $A_{u_1\uparrow}, A_{u_2\uparrow}, \ldots, A_{u_{i-1}\uparrow}$.

  For $i = 1$, we have $\mathrm{Succ}(u\downarrow) = u_1\downarrow$, hence X_{u_1} is $X_u \backslash A_{u\downarrow}$. Given $i \in [1, k-1]$ we have $\mathrm{Succ}(u_i\downarrow) = u_{i+1}$ hence $X_{u_{i+1}} = X_{u_i} \backslash A_{u_i\uparrow}$. This directly yields together with the induction hypothesis at i the result at $i + 1$, and we conclude by induction that $X \backslash X_{u_k}$ is the disjoint union of $A_{u\downarrow}$ and of $A_{u_i\uparrow}$ for all $i \in [1, k]$.

  Finally we have $\mathrm{Succ}(u_k\uparrow) = u\uparrow$ hence $Y_u = X_{u_k} \backslash A_{u_k\uparrow}$ which concludes. □

Attractor-labellings and edges. We say that an edge $e = v \xrightarrow{q} v'$ in G is *not present at* u if $v \in X$ but e does not belong to $G_X^{\leqslant p}$: either $v' \notin X$ or $q > p$. We extend this terminology to positions by considering the associated nodes. We say that e is *removed at position* x if it belongs to $G_X^{\leqslant p}$ but is not present at $\mathrm{Succ}(x)$.

Lemma 11.3 (Characterizing removed edges)

Let $x \in \overline{T}^{\uparrow\downarrow}$ be a position, let $e = v \xrightarrow{q} v'$ be an edge in G and assume that the node u associated to x is not a leaf.

- If $x = u\uparrow$, then e is removed at x if and only if $v \in (X \backslash A) \cap V_{\mathrm{P}}$, $q \leqslant p$ and $v' \in A$.

- Otherwise, e is removed at x if and only if $v \in (X \backslash A) \cap V_{\overline{\mathrm{P}}}$, $q \leqslant p$ and either $q = p$ or $v' \in A$.

Proof. Let $x' = \mathrm{Succ}(x)$ and recall that by definition we have $X' = X \backslash A$. We have in the first case $p' \geqslant p$, and in the second, since u is not a leaf, $p' = p - 1$. We start with the converse implications which are direct. If $v \in X \backslash A$, $q \leqslant p$ and $v' \in A \subseteq X$ then e belongs to $G_X^{\leqslant p}$ and $v \in X'$ but $v' \notin X'$ hence e is removed at x. Now if $x = u{\downarrow}$ and $q = p$, then the assumption $v' \in A$ is no longer required since $p' = q - 1 < q$.

Assume now that e is removed at x. Then $v \in X' = X \backslash A$, and e belongs to $G_X^{\leqslant p}$ so it must be that $q \leqslant p$.

- If $x = u{\uparrow}$ then $p' \geqslant p$, hence it must be that $v' \notin X'$, but since $v' \in X$ this yields $v' \in A = \mathrm{Attr}^{\overline{\mathrm{P}}}_{G_X^{\leqslant p}}(Y_u)$. Since A is a $\overline{\mathrm{P}}$-attractor and $v \notin A$ although v has an edge, namely e, towards A in $G_X^{\leqslant p}$, it must be that $v \in V_{\mathrm{P}}$ which proves our claim.
- If $x = u{\downarrow}$ then $p' = p - 1$ so either $v' \in A = \mathrm{Attr}^{\mathrm{P}}_{G_X^{\leqslant p}}(E_X^p)$, or $q = p$ implying $e \in E_X^p$. In both cases, this implies that $v \in V_{\overline{\mathrm{P}}}$ since $v \notin A$. □

Understanding where the edges that are not present have been removed turns out to be crucial for our needs.

Lemma 11.4 (Properties of removed edges)

If e is not present at u then it is either removed at $u'{\downarrow}$ such that u is below u', or it is removed at $u'{\uparrow}$ such that u is below a strict right sibling of u'.

Figure 11.5 illustrates the situation in the proof.

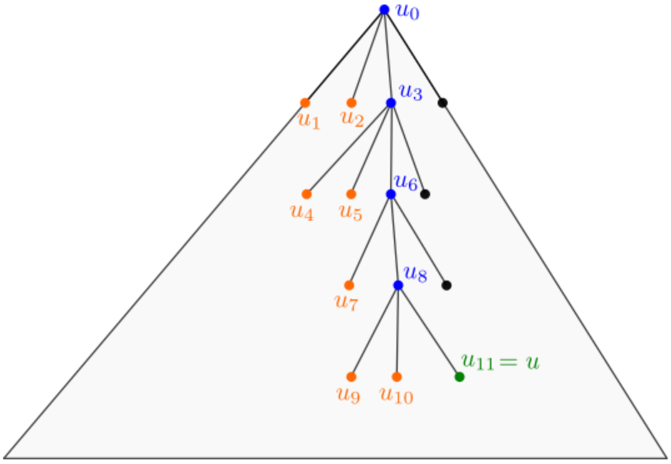


Figure 11.5: An illustration for the proof of Lemma 11.4.

Proof. Let $e = v \xrightarrow{q} v'$. We let $u_0 = \varepsilon$ and for all i such that $u_i \neq u$ we let u_{i+1} be the leftmost child of u_i if u is below u_i, and the direct right sibling of u_i otherwise. It is easy to see that this sequence converges to u, and that for all i, $u_i{\downarrow}$ is before $u{\downarrow}$. Note that all edges (and e in particular) belong to $G_{X_0}^{\leqslant p_0}$ since $X_0 = V$ and $p_0 = d + 1$.

By the first two items of Lemma 11.2, a direct induction concludes that for all i we have $X_{u_i} \supseteq X_u$. In particular since $v \in X_u$ we have for all i, $v \in X_{u_i}$. Since moreover the height of u_i cannot increase with i, there is a unique index i_0 (which is the first index such that $v' \notin X_{u_i}$ if $v' \in X_u$, or such that $p_i < q$ otherwise) such that e belongs to the game associated to u_{i_0} but not to u_{i_0+1}.

If u is below u_{i_0} then e is removed at $u{\downarrow}$ which proves our claim. Otherwise e is removed at $u_{i_0}{\uparrow}$, and u is below a strict right sibling of u_{i_0}. □

2 Simulation by value iteration

We now show that the attractor-based iteration defined in the first section can be simulated by running in parallel two global value iterations, one for each player. In particular this gives an alternative proof of correctness of the algorithm (Theorem 11.1) which is the main result of [JM20].

2.1 Setting

Recall that we have fixed an even integer d and two trees T^{even} and T^{odd} of size $d/2$, respectively interpreted as counting occurrences of even and odd priorities. We let $\mathcal{L}^{\text{odd}}$ be the monotonic graph from Chapter 5 for the (even) parity condition over T^{odd}; it is given by

$$u^{\text{odd}} \xrightarrow{p} u'^{\text{odd}} \text{ in } \mathcal{L}^{\text{odd}} \quad \iff \quad \begin{array}{lll} p \text{ is even and} & u^{\text{odd}} \geqslant_{p+1} u'^{\text{odd}} & \text{or} \\ p \text{ is odd and} & u^{\text{odd}} >_p u'^{\text{odd}}. & \end{array}$$

Likewise, we let $\mathcal{L}^{\text{even}}$ be the monotonic graph over T^{even} given by

$$u^{\text{even}} \xrightarrow{p} u'^{\text{even}} \text{ in } \mathcal{L}^{\text{even}} \quad \iff \quad \begin{array}{lll} p \text{ is odd and} & u^{\text{even}} \geqslant_{p+1} u'^{\text{even}} & \text{or} \\ p \text{ is even and} & u^{\text{even}} >_p u'^{\text{even}}. & \end{array}$$

We also recall the useful notations

$$\lceil p \rceil^{\text{odd}} = \begin{cases} p+1 & \text{if } p \text{ is even} \\ p & \text{if } p \text{ is odd} \end{cases} \quad \text{and} \quad \lceil p \rceil^{\text{even}} = \begin{cases} p+1 & \text{if } p \text{ is odd} \\ p & \text{if } p \text{ is even.} \end{cases}$$

By a slight abuse which is convenient, we often identify Eve with even and Adam with odd, for instance we use the notations u^{P} or $\lceil u \rceil^{\mathrm{P}}$ when $\mathrm{P} \in \{\text{Eve}, \text{Adam}\}$. Note that we have

$$u^{\mathrm{P}} \xrightarrow{p} u'^{\mathrm{P}} \text{ in } \mathcal{L}^{\mathrm{P}} \quad \Longrightarrow \quad u^{\mathrm{P}} \geqslant_{\lceil p \rceil^{\mathrm{P}}} u'^{\mathrm{P}},$$

with a strict inequality if $\lceil p \rceil^{\mathrm{P}} = p$, which means that the parity of p matches P. We also recall that the preorders are defined (even over tuples of non-matching size) by first truncating up to the indicated index and then comparing lexicographically.

For clarity we denote by $\top^{\text{even}}$ the maximal element of the completion $\mathcal{L}^{\text{even},\top}$ of $\mathcal{L}^{\text{even}}$ and use $T^{\text{even},\top} = T^{\text{even}} \cup \{\top^{\text{even}}\}$ for its set of vertices, and likewise for odd. We also let $\mathrm{Upd}^{\mathrm{P}}$ denote $\mathrm{Upd}_G^{\mathcal{L}^{\mathrm{P},\top}}$ for both P. Note here that $\mathrm{Upd}^{\text{odd}}$ is defined as usually, but we dualise the role of the players in $\mathrm{Upd}^{\text{even}}$. In general, these are defined by

$$\mathrm{Upd}^{\mathrm{P}}(\phi^{\mathrm{P}})(v) = \begin{cases} \min\limits_{v \xrightarrow{p} v' \text{ in } G} \rho^{\mathrm{P}}(\phi^{\mathrm{P}}(v'), p) & \text{if } v \in V_{\bar{\mathrm{P}}} \\ \max\limits_{v \xrightarrow{p} v' \text{ in } G} \rho^{\mathrm{P}}(\phi^{\mathrm{P}}(v'), p) & \text{if } v \in V_{\mathrm{P}}, \end{cases}$$

where ρ^{P} denotes the min-predecessor table in $\mathcal{L}^{\mathrm{P},\top}$.

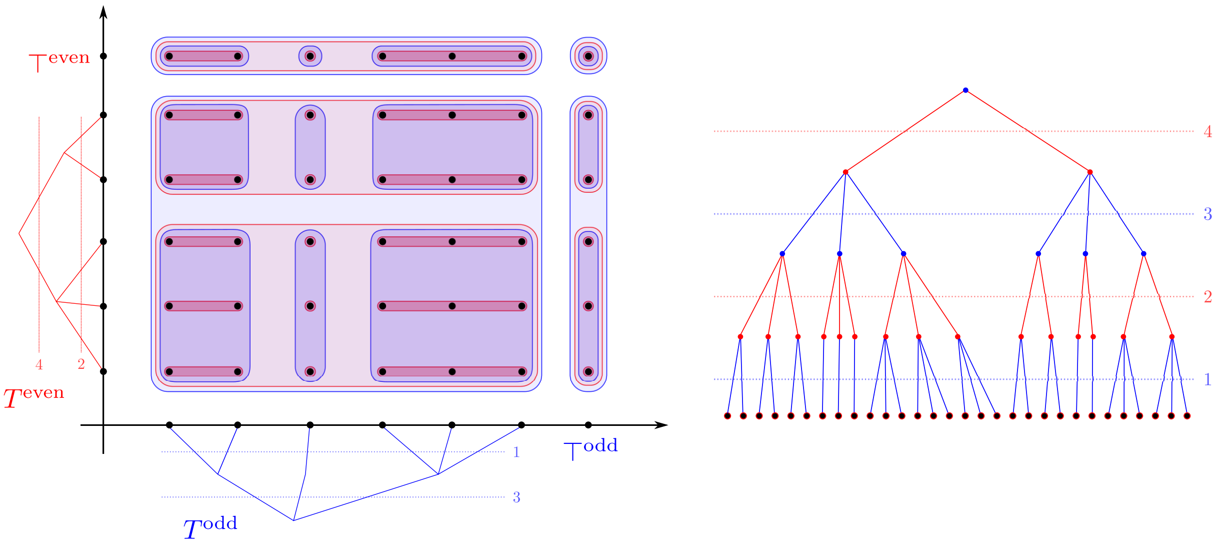


Figure 11.6: An example with $d = 4$. On the left, a representation of the grid $T^{\mathrm{odd},\top} \times T^{\mathrm{even},\top}$, given by the $(6+1) \times (5+1)$ black points. On the right, the interleaving T of the two trees and the nodes of $\overline{T}$. Each node $u' \in \overline{T}$ corresponds to an "inner box" (of corresponding colour) in the grid, comprised of points $(u^{\mathrm{even}}, u^{\mathrm{odd}})$ such that u is below u'. A pair of progress measures $(\phi^{\mathrm{even}}, \phi^{\mathrm{odd}})$ corresponds to a point in the grid for each vertex v.

2.2 Simulation

Given a position $x \in \overline{T}^{\uparrow\downarrow}$, we say that it is *even* if it computes an Eve attractor, and *odd* otherwise. Stated differently, $u\!\downarrow$ is even if and only if u is even, and $u\!\uparrow$ is even if and only if u is odd.

We say that the *complexity at position* x is the attractor-depth of A_x, which we denote r_x. We define the even and odd complexities *strictly before* x by

$$r^{\mathrm{even}}_{<x} = \sum_{\substack{y<x \\ y \text{ even}}} r_y \qquad \text{and} \qquad r^{\mathrm{odd}}_{<x} = \sum_{\substack{y<x \\ y \text{ odd}}} r_y,$$

and the even and odd complexities before x, denoted $r^{\mathrm{even}}_{\preccurlyeq x}$ and $r^{\mathrm{odd}}_{\preccurlyeq x}$ are defined likewise.

For both P we let ϕ^{P}_0 denote the minimal progress measure in $\mathcal{L}^{\mathrm{P},\top}$. Given a position x, we let ϕ^{P}_x be the progress measure obtained after as many global iterations as the sum of attractor-depths of P-attractors computed (strictly) before x, formally

$$\phi^{\mathrm{P}}_x = (\mathrm{Upd}^{\mathrm{P}})^{r^{\mathrm{P}}_{<x}}(\phi^{\mathrm{P}}_0).$$

Recall that $\varepsilon\!\uparrow \in \overline{T}^{\uparrow\downarrow}$ is the greatest position, therefore the even and odd complexities before $\varepsilon\!\uparrow$ respectively correspond to the sum of depths of all even and odd attractors computed throughout the iteration. We also recall that W denotes the output $A_{\varepsilon\uparrow}$ of the iteration. Our main result is the following.

Theorem 11.2 (Separation via simulation)

For all $v \in W$ we have $\phi^{\mathrm{even}}_{\varepsilon\uparrow}(v) = \top^{\mathrm{even}}$ and for all $v \notin W$ we have $\phi^{\mathrm{odd}}_{\varepsilon\uparrow}(v) = \top^{\mathrm{odd}}$.

Stated differently, performing a step of global value iteration (for either player, depending on the attractor) for each attractor-layer computed throughout the iteration results in a pair of progress measures which has "converged" in a weak sense: any vertex is mapped to $\top$ in (at least) one of the two progress measures. Note that this does not imply that $\phi^{\mathrm{P}}_{\varepsilon\uparrow}$ coincides with the least fixpoint ψ^{P}.

However, if both trees are universal then this is sufficient for declaring that W is the winning region of Eve since a vertex mapped to $\top^{\mathrm{odd}}$ is necessarily winning for Adam[3] (see Theorem 4.3) and vice-versa. Therefore Theorem 11.2 implies Theorem 11.1.

The proof of the theorem amounts to a very careful bookkeeping of the evolution of the two progress measures. We break it into inductive properties which we call *invariants* and introduce now.

Invariants. We think of ϕ^{P}_x as the progress measures "just before x". We will prove the following invariants. In full, the names of the invariants read as "before, down", "after, down", "before, up" and "after, up". We let $x' = \mathrm{Succ}(x) \in \overline{T}^{\uparrow\downarrow}$.

(BD) If $x = u\!\downarrow$ then *for both* P, and for all $x \in X$ we have $\phi^{\mathrm{P}}_x(v) \geqslant_{\lceil p\rceil^{\mathrm{P}}} u^{\mathrm{P}}$. In the three invariants below P denotes the parity of u, which is by definition the parity of its height p.

(AD) If $x = u\!\downarrow$ then for all $v \in A_x$ we have $\phi^{\mathrm{P}}_{x'}(v) \geqslant_p u^{\mathrm{P}}$.

(BU) If $x = u\!\uparrow$ we have

$(BU1)$ for all $v \in X \backslash Y$ it holds that $\phi^{\mathrm{P}}_x(v) >_p u^{\mathrm{P}}$ and

$(BU2)$ for all $v \in Y$ it holds that $\phi^{\overline{\mathrm{P}}}_x(v) >_{p+1} u^{\overline{\mathrm{P}}}$. (Note that $\lceil p\rceil^{\overline{\mathrm{P}}} = p+1$.)

(AU) If $x = u\!\uparrow$ then for all $v \in A_x$ we have $\phi^{\overline{\mathrm{P}}}_{x'}(v) >_{p+1} u^{\overline{\mathrm{P}}}$.

We say that *all invariants hold up to* x if for each $y \prec x$ both invariants (before and after y) hold, and moreover the invariant before x (either (BU) or (BD) according to the orientation of x) holds. Note that we have $\phi^{\mathrm{P}}_y \leqslant \phi^{\mathrm{P}}_x$ whenever $y \prec x$ which is very convenient for the proofs below: to prove that all invariants are satisfied up to $x' = \mathrm{Succ}(x)$ it suffices to verify the invariants after x and before x'.

We break the proof into two different but analogous cases according to the orientation of x.

Lemma 11.5 (Descending lemma)

If $x = u\!\downarrow$ and all invariants hold up to x then all invariants hold up to $\mathrm{Succ}(x)$.

Proof. Recall that we have in this case $A = \mathrm{Attr}^{\mathrm{P}}_{G^{\leqslant p}_X}(E^p_X)$ and that $u = u_d u_{d-1} \dots u_p$ is indexed with integers up to p, whose parity is P. Let $A_0, \dots, A_{r-1}$ denote the attractor layers; we have $r = r^{\mathrm{P}}_x$. Locally to the proof we let ϕ^{P}_0 denote ϕ^{P}_x and $\phi^{\mathrm{P}}_{i+1} = \mathrm{Upd}^{\mathrm{P}}(\phi^{\mathrm{P}}_i)$ for $i \in r$, note that by definition $\phi^{\mathrm{P}}_{x'} = \phi^{\mathrm{P}}_r$ and $\phi^{\overline{\mathrm{P}}}_{x'} = \phi^{\overline{\mathrm{P}}}_x$. Note also that the ϕ^{P}_i's satisfy all invariants up to x since they are greater than ϕ^{P}_x. We have to prove the invariant after x and before x' and we start with the former.

We prove by induction on $i \in r$ that for all $v \in A_i$ we have $\phi^{\mathrm{P}}_{i+1}(v) \geqslant_p u^{\mathrm{P}}$. We prove together the base and inductive cases. We let $v \in A_i$ and put $w^{\mathrm{P}} = \phi^{\mathrm{P}}_{i+1}(v) = \mathrm{Upd}^{\mathrm{P}}(\phi^{\mathrm{P}}_i)(v)$.

[3] This is not a typo, recall that $\mathcal{L}^{\mathrm{odd}}$ is universal for the even-parity condition.

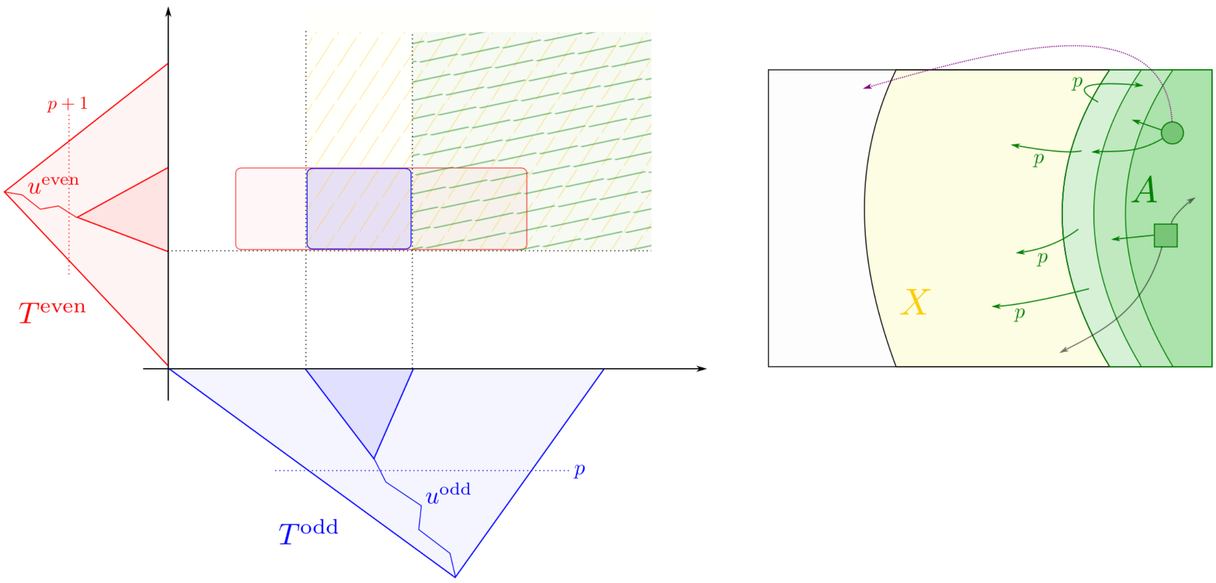


Figure 11.7: Depiction for the descending lemma in the case where P is odd. The invariant before $x = u{\downarrow}$ is depicted in yellow: all vertices belonging to X (on the right) are mapped to the yellow zone (on the left). The invariant after x is depicted in green: we must verify that all vertices in A are mapped by $\phi_{x'}$ to the green zone. The tricky part of the proof is about dealing with $v \in A \cap V_{\overline{\mathrm{P}}}$, represented as the green circle. We must in this case handle edges such as the purple one, which were removed in previous steps.

- Assume first $v \in A_i \cap V_{\mathrm{P}}$. If $i = 0$ then v has a p-edge towards $v' \in X$. By definition of Upd^P it must be that $w^{\mathrm{P}} \xrightarrow{p} \phi_i^{\mathrm{P}}(v')$ in $\mathcal{L}^{\mathrm{P}}$ which implies $w^{\mathrm{P}} >_p \phi_i^{\mathrm{P}}(v')$. Since the invariant before $x = u{\downarrow}$ holds we have $\phi_i^{\mathrm{P}}(v') \geqslant_p u^{\mathrm{P}}$ therefore $w^{\mathrm{P}} >_p u^{\mathrm{P}}$.

  If $i > 0$ then v has an edge $v \xrightarrow{q} v'$ towards $v' \in A_{i-1}$. Again $w^{\mathrm{P}} \xrightarrow{q} \phi_i^{\mathrm{P}}$ in $\mathcal{L}^{\mathrm{P}}$ which implies $w^{\mathrm{P}} \geqslant_{\lceil q\rceil^{\mathrm{P}}} \phi_i^{\mathrm{P}} >_p u^{\mathrm{P}}$ where the second inequality holds by induction. This implies $w^{\mathrm{P}} >_p u^{\mathrm{P}}$ since $\lceil q\rceil^{\mathrm{P}} \leqslant p$ (which follows from the fact that $\lceil p\rceil^{\mathrm{P}} = p$.)

- Assume now that $v \in A_i \cap V_{\overline{\mathrm{P}}}$; by definition of $\mathrm{Upd}^{\mathrm{P}}$ there is an edge $e = v \xrightarrow{q} v'$ in G such that $w^{\mathrm{P}} \xrightarrow{q} \phi_i^{\mathrm{P}}(v')$ in $\mathcal{L}^{\mathrm{P}}$. Since $v \in A_i$ if e belongs to $G_X^{\leqslant p}$ then $v' \in X$ and either $q = p$ or $v' \in A_j$ for some $j < i$. Otherwise, the edge e is not present at u and we will invoke the previous lemmas and the invariant relative to the position where e is removed. We now elaborate on the three cases.

  - Assume $v' \in X$ and $q = p$. Then we have $w^{\mathrm{P}} >_p \phi_i^{\mathrm{P}}(v')$ which implies the wanted result since we have $\phi_i^{\mathrm{P}}(v') \geqslant_p u^{\mathrm{P}}$ thanks to the invariant before x.
  - Assume $v' \in A_j$ for some $j < i$. Then we have $w^{\mathrm{P}} \geqslant_{\lceil q\rceil^{\mathrm{P}}} \phi_i^{\mathrm{P}}(v')$ which is $>_p u^{\mathrm{P}}$ by induction and we conclude as previously.
  - Assume finally that e is not present at u. Then Lemma 11.4 provides us with two possible cases.
    - $\ast$ If e is removed at $u'{\downarrow}$ such that u is below u' (in particular, u cannot be a leaf since $u' \neq u$). Then by Lemma 11.3 we have $v \in V_{\overline{\mathrm{P}'}}$ hence $\mathrm{P} = \mathrm{P}'$, $q \leqslant p'$ and either $v \in A'$ or $q = p'$. In the first case, the invariant after $u'{\downarrow}$ yields $\phi_i^{\mathrm{P}}(v') >_p u'^{\mathrm{P}}$, implying the result since $w^{\mathrm{P}} \geqslant_{\lceil q\rceil^{\mathrm{P}}} \phi_i^{\mathrm{P}}(v')$ and $q \leqslant p'$ (hence $\lceil q\rceil^{\mathrm{P}} \leqslant p'$) and moreover u'^{P} is a prefix of u^{P}.

In the second case we have $w^{\mathrm{P}} >_{p'} \phi_i^{\mathrm{P}}(v')$ which is $\geqslant_p u'^{\mathrm{P}}$ thanks to the invariant before $u'\!\downarrow$, implying that $w^{\mathrm{P}} >_p u'^{\mathrm{P}}$ which is again stronger than the wanted result.

∗ If e is removed at $u'\!\uparrow$ where u is below a strict right sibling of u' then by Lemma 11.3 we have $v \in V_{\overline{\mathrm{P}}'}$ hence $\mathrm{P}' = \overline{\mathrm{P}}$, $q \leqslant p'$ and $v' \in A'$. Thanks to the invariant after $u'\!\uparrow$ we have $\phi_x^{\overline{\mathrm{P}}'}(v') >_{p'+1} u'^{\overline{\mathrm{P}}'}$ and therefore $w^{\mathrm{P}} \geqslant_{\lceil q \rceil^{\mathrm{P}}} \phi_i^{\mathrm{P}}(v') >_{p'+1} u'^{\mathrm{P}}$. Since $\lceil q \rceil^{\mathrm{P}} \leqslant p'+1$ this implies $w^{\mathrm{P}} >_{p'+1} u'^{\mathrm{P}}$ which is equal to u''_{P} where u'' is the parent of u' since the height of u' has the inverse parity of P, implying the result since u is below u''.

This concludes the proof of the induction, and therefore the invariant after x holds. There remains to prove the invariant before x', whose definition depends on the orientation of x'.

- If x' is ascending then since $x = u\!\downarrow$ it must be that u is a leaf and $x' = u\!\uparrow$. In this case, $G_X^{\leqslant p}$ is comprised only of 1-edges therefore we have $A = X$ and $Y' = X \backslash A = \varnothing$. Hence there is only to prove that for all $v \in X \backslash Y = X = A$, $\phi_{x'}^{\mathrm{P}}(v') >_p u^{\mathrm{P}}$, which was done above.

- If x' is descending then $x' = u'\!\downarrow$ where u' is the first child of u. Then invariant before x gives for both players P″ and for all $v \in X' \backslash X$ the inequality $\phi_{x'}^{\mathrm{P}''}(v) \geqslant_{\lceil p \rceil^{\mathrm{P}''}} u^{\mathrm{P}''}$. Since by definition $\phi_{x'}^{\mathrm{P}''}(v)$ is a leaf and since u' is the first child of u this implies $\phi_{x'}^{\mathrm{P}''}(v) \geqslant_{\lceil p' \rceil^{\mathrm{P}''}} u'^{\mathrm{P}''}$. □

Lemma 11.6 (Ascending lemma)

If $x = u\!\uparrow$ and all invariants hold up to x then all invariants hold up to $\mathrm{Succ}(x)$.

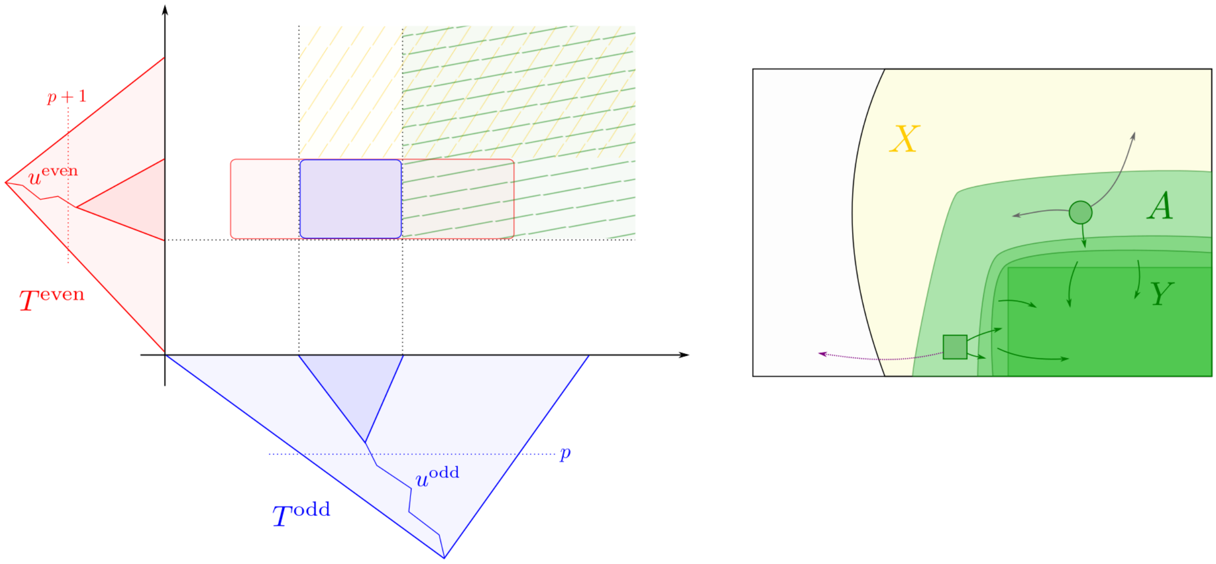


Figure 11.8: Depiction for the ascending lemma in the case where P is odd. The invariant before $u = x\!\uparrow$ states that vertices in $X \backslash Y$ are mapped to the yellow zone, and vertices in Y are mapped to the green zone. We aim to prove that vertices in A are mapped by $\phi_{x'}$ to the green zone as well. This time the harder case corresponds to $v \in A \cap V_{\mathrm{P}}$ (informally, it is always the player who does not control the attractor).

The first part of the proof (verifying the invariant after x) is very similar to the above one, with a focus rather on the other player and a vertex-attractor. The second part however requires a different case analysis.

Proof. Recall that we now have $A = \mathrm{Attr}^{\overline{\mathrm{P}}}_{G_X^{\leqslant p}}(Y)$, and again $u = u_d u_{d-1} \dots u_p$, the parity of p being P. We let $A_0, \dots, A_{r-1}$ denote the attractor layers with depth $r-1$ (recall our convention for vertex-attractors), and by definition we have $A_0 = Y$. We let $\phi_0^{\overline{\mathrm{P}}} = \phi_x^{\overline{\mathrm{P}}}$ and for $i \in r-1$ we let $\phi_{i+1}^{\overline{\mathrm{P}}} = \mathrm{Upd}^{\overline{\mathrm{P}}}(\phi_i^{\overline{\mathrm{P}}})$. The progress measures at x' are given by $\phi_{x'}^{\mathrm{P}} = \phi_x^{\mathrm{P}}$ and $\phi_{x'}^{\overline{\mathrm{P}}} = \phi_{r-1}^{\overline{\mathrm{P}}}$. We start by showing the invariant after x.

We prove by induction on $i \in r$ that for all $v \in A_i$ we have $\phi_i^{\overline{\mathrm{P}}}(v) >_{p+1} u^{\overline{\mathrm{P}}}$. The base case $i = 0$ is exactly given by the second property of the invariant before x. We let $i \in r-1$, pick $v \in A_{i+1}$ and let $w^{\overline{\mathrm{P}}} = \phi_{i+1}^{\overline{\mathrm{P}}}(v) = \mathrm{Upd}^{\overline{\mathrm{P}}}(\phi_i)(v)$, which we aim to prove to be $>_{p+1} u^{\overline{\mathrm{P}}}$.

- Assume first that $v \in A_{i+1} \cap V_{\overline{\mathrm{P}}}$. By definition of the attractor there is an edge $v \xrightarrow{q} v'$ towards $v' \in A_i$ in $G_X^{\leqslant p}$ therefore $q \leqslant p$. By definition of $\mathrm{Upd}^{\overline{\mathrm{P}}}$ it holds that $w^{\overline{\mathrm{P}}} \xrightarrow{q} \phi_i^{\overline{\mathrm{P}}}(v')$ belongs to $\mathcal{L}^{\overline{\mathrm{P}}}$ thus $w^{\overline{\mathrm{P}}} \geqslant_{\lceil q \rceil^{\overline{\mathrm{P}}}} \phi_i^{\overline{\mathrm{P}}}(v') >_{p+1} u^{\overline{\mathrm{P}}}$, where the second inequality holds by induction hypothesis. This yields the wanted result since $\lceil q \rceil^{\overline{\mathrm{P}}} \leqslant p+1$.

- Assume now that $v \in A_{i+1} \cap V_{\mathrm{P}}$. By definition of $\mathrm{Upd}^{\overline{\mathrm{P}}}$ there is an edge $e = v \xrightarrow{q} v'$ in G such that $w^{\overline{\mathrm{P}}} \xrightarrow{q} \phi_i^{\overline{\mathrm{P}}}(v')$ belongs to $\mathcal{L}^{\overline{\mathrm{P}}}$. If e belongs to $G_X^{\leqslant p}$ then $q \leqslant p$ and since $v \in A_{i+1}$ it holds that $v' \in A_{\leqslant i}$ and we conclude as in the first item. Ohterwise e is not present at x and Lemma 11.4 provides us with two cases.

  - If e is removed at $u'\!\downarrow$ such that u is below u' then Lemma 11.3 tells us that $v \in V_{\overline{\mathrm{P}'}}$ hence $\mathrm{P}' = \overline{\mathrm{P}}$ and either $v' \in A'$ or $q = p'$. In the first case the invariant after $u'\!\downarrow$ gives $\phi_i^{\mathrm{P}'}(v') >_{p'} u'^{\mathrm{P}'}$ and moreover $w^{\mathrm{P}'} >_{\lceil q \rceil^{\mathrm{P}'}} \phi_i^{\mathrm{P}'}(v')$ which implies $w^{\mathrm{P}'} >_{p'} u'^{\mathrm{P}'}$ and yields the wanted result since $\mathrm{P}' = \overline{\mathrm{P}}$ and u is below u'.

    In the second case we have $w^{\mathrm{P}'} >_{p'} \phi_i^{\mathrm{P}'}(v')$ which is $\geqslant_{p'} u''^{\mathrm{P}'}$ thanks to the invariant before $u'\!\downarrow$, implying that $w^{\mathrm{P}'} >_{p'} u'^{\mathrm{P}'}$ and the wanted result.

  - If e is removed at $u'\!\uparrow$ such that u is below a strict right sibling of u' then Lemma 11.3 gives $v \in V_{\mathrm{P}'}$ thus $\mathrm{P} = \mathrm{P}'$, and moreover $q \leqslant p'$ and $v' \in A'$. We have $w^{\overline{\mathrm{P}}} \geqslant_{\lceil q \rceil^{\overline{\mathrm{P}}}} \phi_i^{\overline{\mathrm{P}}}(v')$ and moreover the invariant after $u'\!\uparrow$ gives $\phi_i^{\overline{\mathrm{P}}}(v') >_{p'+1} u'^{\overline{\mathrm{P}}}$ and since $\lceil q \rceil^{\overline{\mathrm{P}}} \leqslant p'+1$ we have $w^{\overline{\mathrm{P}}} >_{p'+1} u'^{\overline{\mathrm{P}}}$. This rewrites as $w^{\overline{\mathrm{P}}} >_{p'+1} u''^{\overline{\mathrm{P}}}$ where u'' is the parent of u' and implies the wanted result since u is below u''.

This concludes the induction and the proof of the invariant after x. We now prove the invariant before x', whose definition depends on the orientation of x'.

- If $x' = u'\!\downarrow$ then u' is the direct right sibling of u. For $\overline{\mathrm{P}}$ we have $u'^{\overline{\mathrm{P}}} = u^{\overline{\mathrm{P}}}$ and the result follows directly from the invariant before $u\!\downarrow$. For P we have for all $v \in X' = X \backslash A \subseteq X \backslash Y$ that $\phi_{x'}^{\mathrm{P}}(v) >_p u^{\mathrm{P}}$ thanks to the invariant before $x = u\!\uparrow$ and the result follows.

- If $x' = u'\!\uparrow$ then u is the rightmost child of u' and we have $Y' = X \backslash A, p' = p+1$ and $\mathrm{P}' = \overline{\mathrm{P}}$. There are two items to prove.

  - Let us show that for all $v \in X' \backslash Y'$ we have $\phi_{x'}^{\mathrm{P}'}(v) >_{p'} u'^{\mathrm{P}'}$. By the third item in Lemma 11.2, $X' \backslash Y'$ is the disjoint union of $A_{u'\downarrow}$ and of $A_{u''\uparrow}$ where u'' ranger over the children of u'. For $v \in A_{u'\downarrow}$ the invariant after $u'\!\downarrow$ concludes. For $v \in A_{u''\uparrow}$ the invariant after $u''\!\uparrow$ yields $\phi_{x'}^{\overline{\mathrm{P}}''}(v) >_{p''+1} u''^{\overline{\mathrm{P}}''}$ which concludes since $\overline{\mathrm{P}}'' = \mathrm{P}'$ and $u''^{\overline{\mathrm{P}}''} = u'^{\mathrm{P}'}$.

- There remains to show that for all $v \in Y' = X\backslash A$ we have $\phi^{\mathrm{P}}_{x'} >_{p'+1} u'^{\mathrm{P}}$. Since $A \supseteq Y$ it holds that $Y' = X\backslash A \subseteq X\backslash Y$ hence by the invariant before $u{\uparrow}$ we have for $v \in Y'$ that $\phi^{\mathrm{P}}_{x'}(v) >_p u^{\mathrm{P}}$. Since u is the rightmost child of u we have $u^{\mathrm{P}} = u'^{\mathrm{P}}k \in \overline{T}$ for k maximal therefore this gives $\phi^{\mathrm{P}}_{x'}(v) >_{p+2} u'^{\mathrm{P}}$, the wanted result. □

Now the invariants before and after $\varepsilon \uparrow$ give exactly the theorem since we have in this case $p = d + 1$ and $W = A_{u\uparrow}$.

3 Accelerating iterations of parity bi-progress measures

Moving on from the algorithm of Jurdziński and Morvan [JM20], we now introduce a systematic way of accelerating synchronous value iterations such as those of the previous section. More precisely, we explore the following question

> "how can we exploit information from ϕ^{even} to accelerate ϕ^{odd}, and vice-versa?"

Our initial inspiration was to simulate Zielonka's algorithm in a value iteration scenario, which seemed to require such an acceleration mechanism.

We fix a finite $[1, d]$-arena G of size n.

3.1 The structure of parity bi-progress measures

We let $\mathcal{L}^{\text{odd}}$ and $\mathcal{L}^{\text{even}}$ be two arbitrary finite monotonic $[1, d]$-graphs respectively satisfying $W = \text{Parity}_{[1,d]}$ and its complement. As before, we use $\top^{\text{odd}}$ and $\top^{\text{even}}$ to denote the maximal elements of the completions and use Upd^{odd} and Upd^{even} for the corresponding operators, with dual semantics.

Bi-progress measures. A *bi-progress measure* (for G in $(\mathcal{L}^{\text{odd}}, \mathcal{L}^{\text{even}})$) is a pair $(\phi^{\text{odd}}, \phi^{\text{even}})$ of progress measures. We say that ϕ^{odd} and ϕ^{even} are the *coordinates* of ϕ.

Given $\mathrm{P} \in \{\text{even}, \text{odd}\}$ we say that an edge or a vertex is *valid for* P or P-*valid* if it is valid in ϕ^{P} and that it is *bi-valid* if it is valid for both P. Recall that $\mathcal{L}^{\text{odd}}$ is looked at from the point of view of Eve (occurrences of odd priorities define usual signatures), therefore an Eve-vertex is odd-valid if and only if it has an odd-valid outgoing edge and vice-versa. We hope that this convention does not cause confusion, we find it to be the easier one to work with. We still identify even with Eve and odd with Adam, for instance $V_{\mathrm{P}} = V_{\text{Eve}}$ if $\mathrm{P} = \text{even}$.

Lemma 11.7 (Vertex bi-validity)

Any bi-valid vertex has a bi-valid outgoing edge.

Proof. If $v \in V_{\mathrm{P}}$ is bi-valid then all its outgoing edges are P-valid and moreover one of them is $\overline{\mathrm{P}}$-valid. □

We say that a bi-progress measure ϕ has *weakly converged* if for all $v \in V$, either $\phi^{\text{odd}}(v) = \top^{\text{odd}}$ or $\phi^{\text{even}}(v) = \top^{\text{even}}$. The following result should be seen as a separation result akin to the fact that a given vertex cannot be winning for both players in general. It is not specific to $W = \text{Parity}$.

Corollary 11.1

The progress measure $\psi = (\psi^{\text{odd}}, \psi^{\text{even}})$ has weakly converged.

Proof. By definition, every vertex is bi-valid in ψ. Starting from v, we may therefore produce an infinite path π comprised only of bi-valid edges thanks to the lemma. Stated differently, $\psi^{\text{odd}}(v)$ and $\psi^{\text{even}}(v)$ both have $\text{col}(\pi)$ as a colouration respectively in $\mathcal{L}^{\text{odd},\top}$ and $\mathcal{L}^{\text{even},\top}$. If $\text{col}(\pi) \in W$ this implies $\psi^{\text{even}}(v) = \top^{\text{even}}$ since $\mathcal{L}^{\text{even}}$ satisfies ${}^{\text{c}}W$, and symmetrically. □

This already suggests a weak acceleration mechanism: if it so happens that the iteration is over for a given player, say $\phi^{\text{even}} = \psi^{\text{even}}$ (stated differently, all vertices are even-valid), then any vertex which is not mapped to $\top^{\text{even}}$ can be accelerated, in ϕ^{odd}, to $\top^{\text{odd}}$ (see Figure 11.9), without breaking the invariant of being $\leqslant \psi$ (on both coordinates).

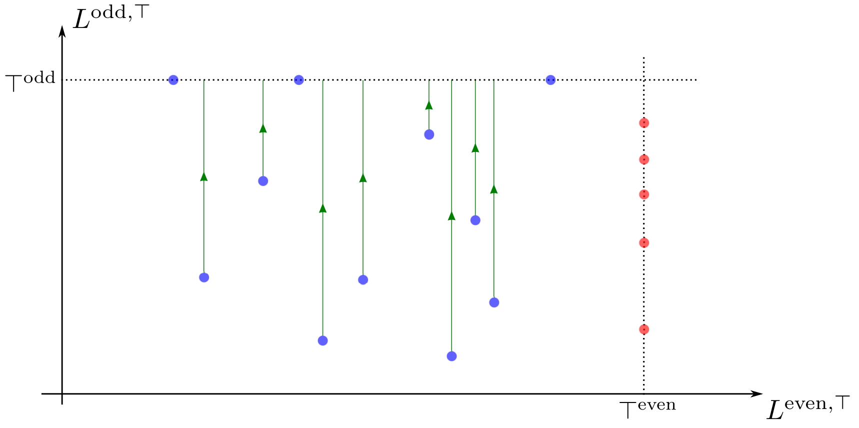


Figure 11.9: The weak acceleration (which is not specific to parity) which can be performed if all vertices are even-valid. Note that the obtained bi-progress measure has weakly converged.

Using the weak acceleration and alternating an iteration in parallel for each players, one obtains a (weakly converging) value iteration algorithm with runtime roughly $\min(|L^{\text{Eve}}|, |L^{\text{Adam}}|)$ rather than roughly $|L^{\text{Eve}}||L^{\text{Adam}}|$. We will show that in the case of the parity condition, one may obtain a local variant of the acceleration mechanism.

Parity bi-progress measures. Recall from Chapter 5 that any graph satisfying the (even) parity condition embeds into $\mathcal{L}_T$, where $T \subseteq \omega^{d/2}$ is the tree comprised of tuples of odd occurrences of vertices of the graph. We assume for the sake of clarity[4] that tuples of odd occurrences in $\mathcal{L}^{\text{odd}}$ are pairwise distinct, in other words the map $L^{\text{odd}} \to T^{\text{odd}}$ is a bijection. Therefore we simply identify vertices in $\mathcal{L}^{\text{odd}}$ with elements of T^{odd}.

Stated differently, we take $\mathcal{L}^{\text{odd}}$ to be a monotonic graph obtained from $\mathcal{L}_{T^{\text{odd}}}$ by (potentially) removing edges. Naturally, we do the same for $\mathcal{L}^{\text{even}}$. Therefore edges in $\mathcal{L}^{\text{P}}$ verify that

$$u^{\text{P}} \xrightarrow{p} u'^{\text{P}} \text{ in } \mathcal{L}^{\text{P}} \quad \Longrightarrow \quad u^{\text{P}} \geqslant_{[p]^{\text{P}}} u'^{\text{P}},$$

with a strict inequality if $[p]^{\text{P}} = p$, which means that the parity of p matches P. We stress the fact that this need not be an equivalence; as we will see there is value in considering non-saturated constructions of monotonic graphs.

[4] This assumption can easily be removed in what follows, it incurs no loss of generality.

We let T denote the interleaving of T^{even} and T^{odd}. Note that Section 2 does not exploit the (interleaved) preorders over T; elements $u, u' \in T$ (or $\overline{T}$ in that case) were only compared via their projections for instance $u^{\text{even}} \geqslant_p u'^{\text{even}}$ for some even p.

What we call the *structure of parity bi-progress measures* is the sequence of preorders induced over the interleaving $T \subseteq \omega^d$. Given $v \in G$ and a bi-progress measure $(\phi^{\text{odd}}, \phi^{\text{even}})$, assuming that $\phi^{\text{odd}}(v) < \top^{\text{odd}}$ and $\phi^{\text{even}}(v) < \top^{\text{even}}$, we define $\phi(v) \in T$ as the interleaving of $\phi^{\text{odd}}(v)$ and $\phi^{\text{even}}(v)$.

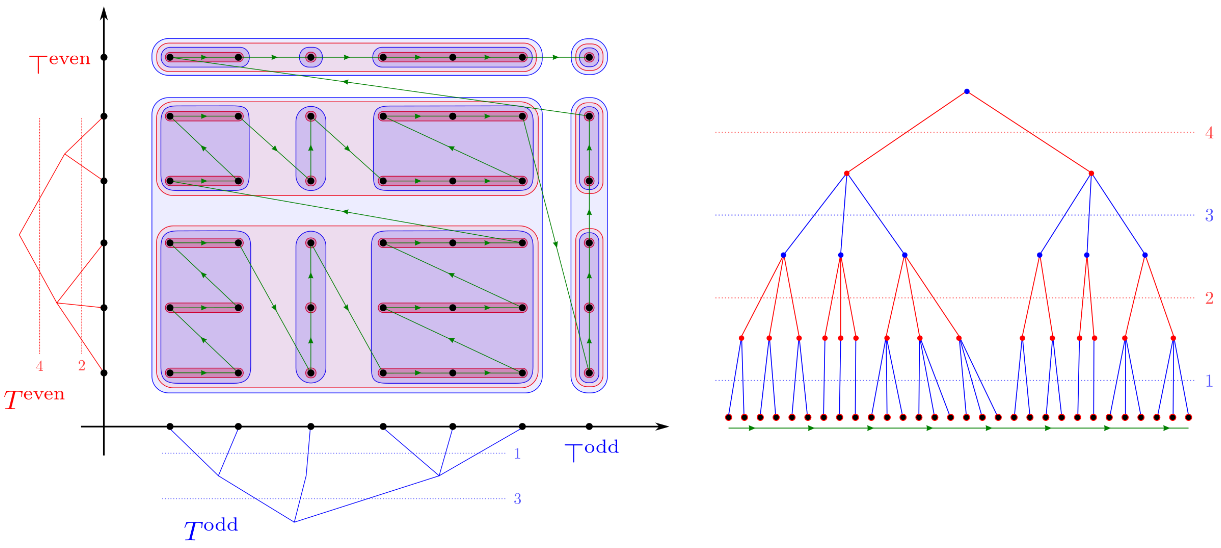


Figure 11.10: Another instance of Figure 11.6, which depicts an example where $d = 4$. The linear order $\geqslant_1$ is depicted in green. The blue and red boxes are now interpreted as equivalence classes $=_p$ respectively for even and odd p's; they are ordered by $\geqslant_p$.

We let $T^{\top}$ denote $(T^{\text{even}} \cup \{\top^{\text{even}}\}) \times (T^{\text{odd}} \cup \{\top^{\text{odd}}\})$ and extend all preorders naturally to $T^{\top}$ by seeing $\top^{\text{odd}}$ as an occurrence of the priority $d+1$ and $\top^{\text{even}}$ as an occurrence of $d+2$. (Formalities are not important and cumbersome, we omit them.)

We raise the reader's attention on the fact that the linear order $\geqslant_1$ over $T^{\top}$ *does not* coincide with the (partial) product order $\geqslant$ over $T^{\top}$ defined via the projections. It is true however that $\geqslant_1$ refines $\geqslant$; we refer to Figure 11.10. We now extend the definition of $\phi(v)$ to all vertices in the obvious way. We therefore see bi-progress measures as mappings $V \to T^{\top}$.

3.2 Accelerations

Here is our main lemma for bi-progress measures.

Lemma 11.8 (Edge bi-validity)

Let $v \xrightarrow{p} v'$ be a bi-valid edge. Then $\phi(v) >_p \phi(v')$.

Proof. Assume p to be even. Then we have $\phi^{\text{even}}(v) >_p \phi^{\text{even}}(v')$ and $\phi^{\text{odd}}(v) \geqslant_{p+1} \phi^{\text{odd}}(v')$ which implies the result. The case of odd priorities is similar. □

We obtain our main result as a consequence. We say that a vertex v is *minimal up to* p or p-*minimal* (in ϕ) if for all vertices v' it holds that $\phi(v') \geqslant_p \phi(v)$. We assume that $\phi \leqslant \psi$ (over both coordinates).

Theorem 11.3 (Acceleration for $\overline{\mathrm{P}}$ over p-minimal vertices)

Let $p \in [1, d]$ and assume that for some P it holds that all p-minimal vertices are P-valid. Then all p-minimal vertices v satisfy $\psi^{\overline{\mathrm{P}}}(v) >_{[p]^{\overline{\mathrm{P}}}} \phi^{\overline{\mathrm{P}}}(v)$.

Stated differently, if the hypothesis of the theorem is met, one may update for all p-minimal vertex the value of $\phi^{\overline{\mathrm{P}}}(v)$ to the smallest $>_{[p]^{\overline{\mathrm{P}}}}$ position (which does not depend on v), without breaking the property of being $\leqslant \psi^{\overline{\mathrm{P}}}$.

Note that unless ϕ has converged, $d+1$-minimal vertices are exactly those which are mapped to T (no $\top$ coordinate). Hence when $p = d+1$ the theorem instantiates as the weak acceleration described in Figure 11.9: if all vertices are valid for P then those which are not mapped to $\top^{\mathrm{P}}$ can be mapped to $\top^{\overline{\mathrm{P}}}$. We refer to Figure 11.11 for a depiction of the acceleration.

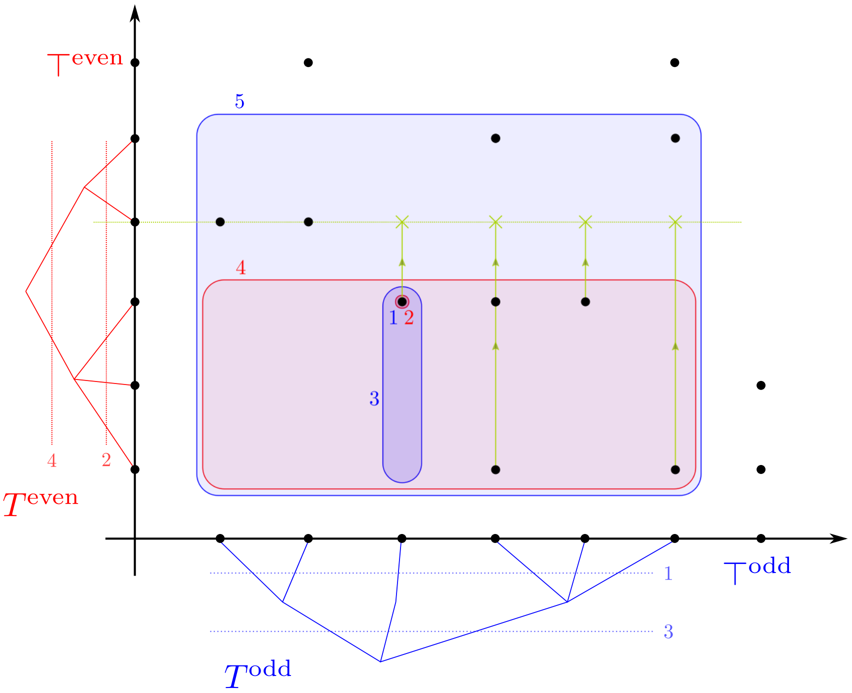


Figure 11.11: An illustration of the acceleration induced by Theorem 11.3. The points on the grid represent positions of the vertices in the current bi-progress measure ϕ. The boxes containing p-minimal vertices for $p \in [1, 5]$ are also displayed; in this case, the boxes are the same for $p = 1$ and 2, and only one vertex is p-minimal for $p \in \{1, 2, 3\}$. Assuming all 4-minimal vertices are valid for Eve (this corresponds to the progress measure ϕ^{odd} displayed horizontally), they made be accelerated in ϕ^{even} as represented by the lime arrows.

Proof. We fix a representative $u_0 \in T^{\top}$ of the $=_p$-equivalence class of p-minimal vertices in ϕ, in other words p-minimal vertices are exactly those which satisfy $\phi(v) =_p u_0$. We let

$$S = \{v \in V \mid \phi(v) =_p u_0 \text{ and } \psi^{\overline{\mathrm{P}}}(v) \leqslant_{[p]^{\overline{\mathrm{P}}}} u_0^{\overline{\mathrm{P}}}\},$$

which we assume for contradiction to be nonempty. We let ϕ' be the bi-progress measure whose coordinates are given by $\phi'^{\mathrm{P}} = \phi^{\mathrm{P}}$ and $\phi'^{\overline{\mathrm{P}}} = \psi^{\overline{\mathrm{P}}}$. We may rewrite membership in S as follows

$$\begin{aligned}
v \in S \quad &\iff \quad \phi(v) =_p u_0 \text{ and } \psi^{\overline{\mathrm{P}}}(v) \leqslant_{[p]^{\overline{\mathrm{P}}}} u_0^{\overline{\mathrm{P}}} \\
&\iff \quad \phi^{\mathrm{P}}(v) =_{[p]^{\mathrm{P}}} u_0^{\mathrm{P}} \text{ and } \phi^{\overline{\mathrm{P}}}(v) =_{[p]^{\overline{\mathrm{P}}}} u_0^{\overline{\mathrm{P}}} \text{ and } \psi^{\overline{\mathrm{P}}}(v) \leqslant_{[p]^{\overline{\mathrm{P}}}} u_0^{\overline{\mathrm{P}}} \\
\text{(since } \phi^{\overline{\mathrm{P}}} \leqslant \psi^{\overline{\mathrm{P}}}\text{)} \quad &\iff \quad \phi^{\mathrm{P}}(v) =_{[p]^{\mathrm{P}}} u_0^{\mathrm{P}} \text{ and } \psi^{\overline{\mathrm{P}}}(v) =_{[p]^{\overline{\mathrm{P}}}} u_0^{\overline{\mathrm{P}}} \\
&\iff \quad \phi'(v) =_p u_0 \\
&\iff \quad \phi'(v) \leqslant_p u_0.
\end{aligned}$$

Now let $v \in S$. It is valid in ϕ^{P} by assumption, and valid in $\psi'^{\overline{\mathrm{P}}}$ since all vertices are valid in evaluations in general; in other words it is bi-valid in ϕ'. Therefore it has a bi-valid outgoing edge $v \xrightarrow{q} v'$ by Lemma 11.7, hence Lemma 11.8 gives

$$\phi'(v) >_q \phi'(v'),$$

which implies $\phi'(v) >_1 \phi'(v')$ since $>_1$ is the finest preorder and therefore

$$u_0 \geqslant_p \phi'(v) \geqslant_p \phi'(v')$$

hence $v' \in S$.

Iterating this process creates an infinite decreasing sequence for $>_1$: a contradiction. Hence S is empty which gives the wanted result. □

4 Simulation of Zielonka's algorithm

We now give a detailed sketch of how Zielonka's algorithm can be obtained as an accelerated value iteration.

4.1 Additional ingredients

In our simulation of Jurdziński and Morvan's algorithm the control we have over positions $\phi(v)$ of vertices (in or out of the current X) is always given by lower bounds (see invariants in Section 2). This gives no information about vertex-validity; actually when performing global iterations as above, we cannot guarantee any such control. Our simulation of Zielonka's algorithm is similar to the one of Section 2, but requires introducing four additional ingredients.

Local lifts and resets. First, we use *local lift* operators, which give us more control over the positions of the vertices (in particular, we no longer want to automatically update vertices outside of the current X). Second, we use *resets* to simulate "discarding information" (as in Figure 2 from the general introduction): we sometimes have to "update back" the value $\phi(v)$ for some $v \in X$ to a smaller value $\phi'(v) \leqslant \phi(v)$ (which is not natural in the context of value iterations).

Accelerations. Third, and most importantly, we use *accelerations* as described in the previous section. These allow to perform "shortcuts" in the attractor-labelling, which are valid (only) if T^{odd} and T^{even} are chosen to be the complete n-ary trees, and can be stated as follows: if it so happens that[5] $Y_u = \varnothing$ at some node u, then it holds that $Y_{u'} = \varnothing$ for any right sibling of u, and therefore we may skip all computation below right-siblings of u and continue instead from the parent of u.

[5] A similar conclusion is also true if $A_{u\uparrow} \setminus Y_u = \varnothing$, which is a weaker assumption. Stated differently, we may also simulate the optimisation of Zielonka's algorithm by Liu, Duan and Tian [LDT14] for free. For simplicity, we stick to Zielonka's algorithm in its simpler form.

Using these shortcuts allows to formalise Zielonka's algorithm as an attractor-labelling and thus apply our simulation technique. Now using an acceleration to simulate a shortcut after $u\uparrow$, provided $Y_u = \varnothing$, requires ensuring that all vertices in X_u are valid for P_u in the progress measure after $u\uparrow$. This will follows from induction (computation below u) for vertices in $X_u \setminus A_{u\downarrow}$, however ensuring that vertices in $A_{u\downarrow}$ are valid for P_u requires introducing a fourth ingredient.

Lazy positions. Fourth, we add (polynomially many) *lazy positions* to the monotonic graph parameterising the value iteration. This corresponds to the (asymmetric) construction from Zielonka's proof [Zie98], and also (although technicalities have to be adapted) to the lazy progress measures from [DJL19], inspired by [Kla91]. Informally, these lazy positions are added to embed vertices in attractors, in such a way that they are guaranteed to remain valid if they should later be accelerated. More details will be given below, we first introduce the construction.

Formally, given a tree $T^{\mathrm{odd}} \subseteq \omega^{d/2}$ whose components are interpreted as odd occurrences, and an integer $n \in \omega$, we let $\mathcal{L}^{\mathrm{lazy},n,\mathrm{odd}}_{T^{\mathrm{odd}}}$ be the monotonic graph over $\overline{T^{\mathrm{odd}}} \times n$ given by

$$(u,i) \xrightarrow{p} (u',i') \qquad \iff \qquad \text{either} \quad \begin{array}{l} p+1 \geqslant p_u \text{ and } p \text{ even } \text{ and } u \geqslant_{p+1} u' \\ p \geqslant p_u \text{ and } p \text{ odd } \text{ and } u >_p u' \\ \lceil p \rceil^{\mathrm{odd}} < p_u \text{ and } u > u' \\ \lceil p \rceil^{\mathrm{odd}} < p_u \text{ and } u = u' \text{ and } i > i', \end{array}$$

where p_u is the (odd) height of u. (Note the striking similarity with the construction for mean-payoff parity games in Chapter 7.) We do not include proofs that $\mathcal{L}^{\mathrm{lazy},n,\mathrm{odd}}_{T^{\mathrm{odd}}}$ is monotonic and satisfies Parity; these facts can verified by hand. The construction is better understood when envisaged in a recursive fashion, this is illustrated in Figure 11.12.

Observe that the restriction of $\mathcal{L}^{\mathrm{lazy},n,\mathrm{odd}}_{T^{\mathrm{odd}}}$ to $\{(u,0) \mid u \in T\}$ corresponds exactly to the usual (odd) signature construction $\mathcal{L}^{\mathrm{odd}}_{T^{\mathrm{odd}}}$. In particular, if T^{odd} is $(n, d/2)$-universal (as a tree), then $\mathcal{L}^{\mathrm{lazy},n,\mathrm{odd}}_{T^{\mathrm{odd}}}$ is Parity-universal. Note that the monotonic graph $\mathcal{L}^{\mathrm{lazy},n,\mathrm{odd}}_{T^{\mathrm{odd}}}$ is not saturated (it is not of the form $\mathcal{L}^{\mathrm{odd}}_{T',\mathrm{odd}}$ for some tree $T'^{,\mathrm{odd}}$), for instance most vertices (all those in $(\overline{T} \setminus T) \times n$) do not have 0-loops.

Of course, we use a dual construction $\mathcal{L}^{\mathrm{lazy},n,\mathrm{even}}_{T^{\mathrm{even}}}$ from the point of view of the opponent Adam; details are again omitted here.

4.2 Detailed description of simulation

To simulate Zielonka's algorithm, we set T^{even} and T^{odd} to be complete n-ary trees of height $d/2$. Actually, the algorithm would remain the same if trees of higher degree are used; we stick with degree n for simplicity. Setting n to be a sufficiently large ordinal (meaning, of cardinal greater than the arena) one may also run the same (accelerated value iteration) algorithm over infinite arenas, which is not surprising considering Zielonka's algorithm is valid over infinite arenas.

Consider an node $u \in \overline{T}$, we describe how the computation performed by the Zielonka algorithm below u is simulated. For convenience, we assume that u is an even node (its height p_u is even), the odd case is of course analogous. We abstain from giving precise invariants, these can easily be inferred from the description below. We refer the reader to Figures 11.13 and 11.14 for helpful depictions of the situation.

Reset. We start by doing a reset: all vertices in X_u are then mapped to $(u^{\mathrm{even}}, 0)$ by ϕ^{even} and to $(u^{\mathrm{odd}}, 0)$ by ϕ^{odd}.

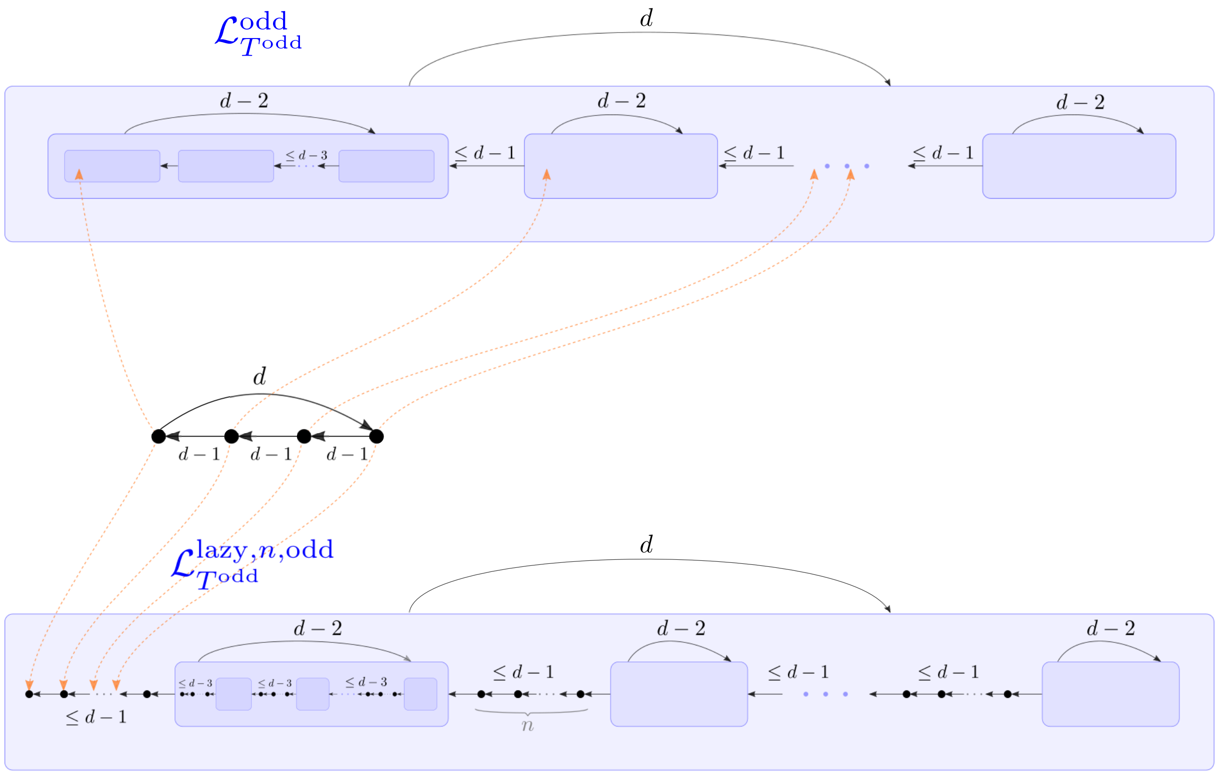


Figure 11.12: At the top, a recursive depiction of the saturated monotonic graph $\mathcal{L}^{\text{odd}}_{\overline{T}^{\text{odd}}}$ corresponding to relevant odd occurrences, defined in Chapter 5. Vertices are not apparent; they correspond to the base case (a single 0-loop), or to the leaves of the tree. We use $\xrightarrow{\leq k}$ to denote the conjunction of $\xrightarrow{\leq i}$ for $i \in [0, k]$, and edges following from left and right composition are not depicted for clarity. In the bottom, the lazy construction $\mathcal{L}^{\text{lazy},n,\text{odd}}_{\overline{T}^{\text{odd}}}$ defined above; here, there are n vertices for each inner node $u \in \overline{T}^{\text{odd}}$ of the tree. Observe that the leftmost node no longer has a $d-2$-loop in the new construction; it is not saturated. In between, a small graph, and its evaluation in both monotonic construction. Intuitively, strategies computed by the lazy graph are “attractor-based”.

Descending at node u. We then proceed to perform local lifts at vertices in X_u until the following holds.

- A vertex v belonging to $A_{u\downarrow}$ (the Eve-attractor in the current arena G_u to edges of even priority p_u) ends up mapped to $(u''^{,\text{even}}, 0)$ by ϕ^{even}, where $u''^{,\text{even}}$ is the $>_{p_u}$-successor[6] of u^{even} in $\overline{T}^{\text{even}}$. By ϕ^{odd}, it ends up mapped to (u^{odd}, i) where i is its attractor-depth; it is then valid in ϕ^{odd} (that is, it is Eve-valid) and remains so provided all vertices in X_u remain mapped to positions below u^{odd} (this remains the case up to $u\uparrow$ when $Y_u = \varnothing$).

- A vertex v belonging to $X_u \backslash A_{u\downarrow}$ ends up mapped to $(u'^{,\text{odd}}, 0)$ by ϕ^{odd}, where $u'^{,\text{odd}}$ is the $>_1$-successor[7] of u^{odd} in $\overline{T}^{\text{odd}}$. It does not move in ϕ^{even}.

Recursive computation below u. We now recursively simulate Zielonka's algorithm below u; we let x' denote $\text{Succ}(u\downarrow)$ and $u' \in \overline{T}$ be the associated node. As a result of the computation below u, $X_{u'} = X_u \backslash A_{u\downarrow}$ is partitioned into the winning regions, Y_u for Adam and $X_{u'} \backslash Y_u$ for Eve, in $G_{u'}$.

[6] This is the direct right sibling of u if it has one, and a “next cousin” otherwise.
[7] If u^{odd} is not a leaf, this is the first child of u^{odd}.

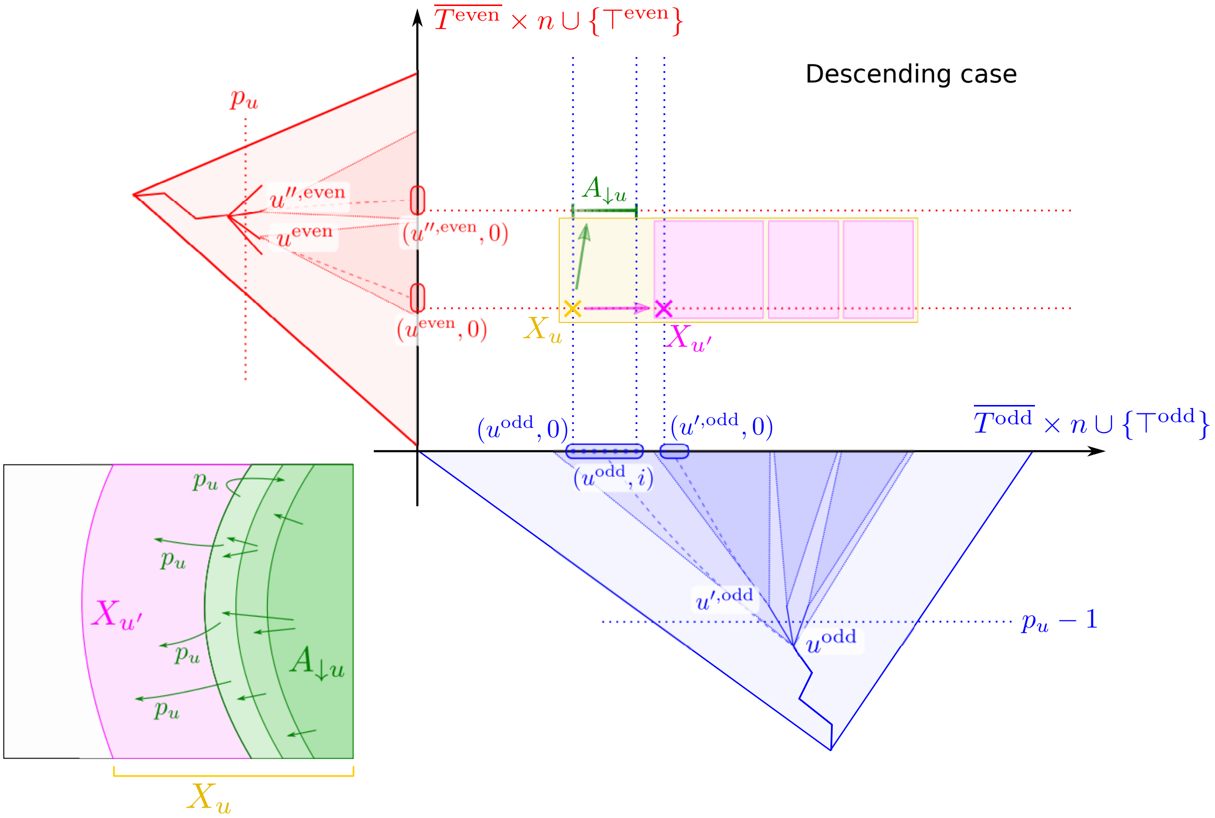


Figure 11.13: A complete depiction for the descending case. The initial reset brings all vertices in X_u to the position marked in yellow. From there, applying polynomially many (at most n^2) local lifts drives vertices in the Eve-attractor A_u to the green zone (according to their attractor layer), and other vertices, in $X_{u'}$, to the position marked in fuchsia. The computation then continues from $u' \downarrow = \iota(u'^{,\text{odd}}, u^{\text{even}}) \downarrow$, which corresponds to the first fuchsia box. The computation below u (or recursive call in Zielonka's algorithm), empties all vertices from the three fuchsia boxes.

It then holds by induction (see Figure 11.14) that vertices in Y_u are mapped by ϕ^{odd} to $(u''^{,\text{odd}}, 0)$ where $u''^{,\text{odd}}$ is the $>_{p_u+1}$-successor of u^{odd}, and they are moreover are valid in ϕ^{odd} (this corresponds to Eve-validity). Likewise, it holds that vertices in $X_{u'} \backslash Y_u$ are mapped by ϕ^{even} to $(u''^{,\text{even}}, 0)$ where $u''^{,\text{even}}$ is the $>_{p_u}$-successor of u^{even}, and they are moreover valid in ϕ^{odd}. Here, the fact that T^{even} and T^{odd} are large enough guarantees that Y_u and its complement are indeed the winning regions, and that the needed validity conditions (described above) indeed hold.

Ascending at node u. For the ascending case, we proceed differently if there is an acceleration (or shortcut) or not.

- Acceleration: assume[8] $Y_u = \varnothing$. All vertices in $X_{u'}$ are valid in ϕ^{odd} (see just above), and since $Y_u = \varnothing$, this is also the case for vertices in A_u (which have been "waiting" in (u^{odd}, i), see descending case). Therefore all $(p_u + 1)$-minimal vertices are valid for Eve, and can be accelerated thanks to Theorem 11.3, that is, mapped to $(u^{(4),\text{even}}, 0)$, where $u^{(4),\text{even}}$ is the $>_{p_u+2}$-successor of u^{even}, by ϕ^{even}. The simulation then resumes from $u''' \uparrow$, where u''' is the parent of u in $\overline{T}$.

[8] As mentioned above, this can also be done with the weaker condition $A_{u\uparrow} = Y_u$, we then obtain the optimisation of [LDT14].

- If there is no acceleration, then local lifts are performed so that vertices in $A_{u\uparrow}$, which is the Adam-attractor to Y_u in G_u, are mapped to $(u^{(5),\text{odd}}, 0)$, where $u^{(5),\text{odd}}$ is the $>_{p_u+1}$-successor of u^{odd} in $\overline{T^{\text{odd}}}$. In ϕ^{even}, a vertex v in $A_{u\uparrow} \backslash Y_u$ ends up in $(u''^{,\text{even}}, i)$, where i is its attractor-depth.

  Notice that all vertices in $A_{u\uparrow} \supseteq Y_u$ are then valid for Adam, and remain so at least until new vertices become mapped to positions $>_{p_u+2} u^{\text{even}}$. This fact is important to guarantee the needed induction hypothesis. The simulation then resumes from the successor of $u\uparrow$.

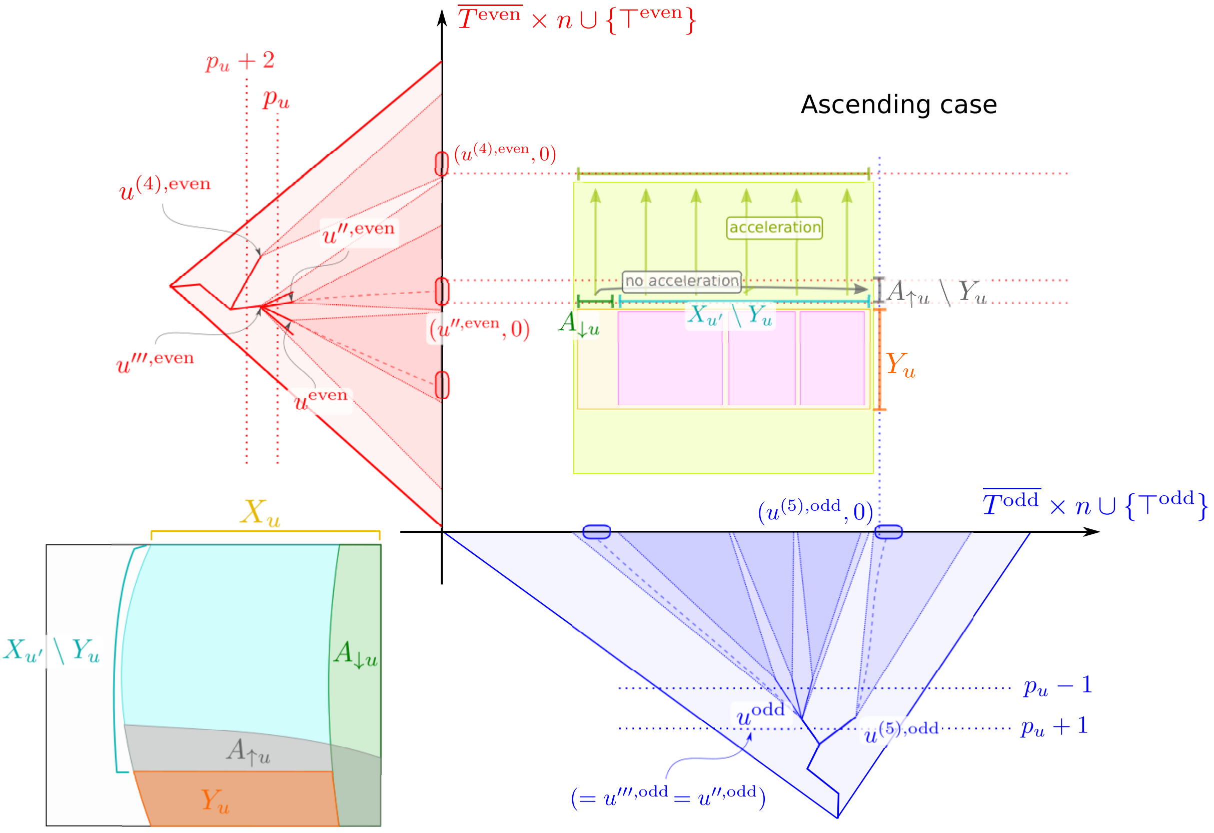


Figure 11.14: A complete depiction for the two cases (accelerating or not) in the ascending case at u. Initially, the vertices in $X_{u'}$ are partitioned between those winning for Adam in $G_{u'}$, which correspond to Y_u and are mapped in the orange zone, and those winning for Eve, which are mapped in the cyan zone. If $Y_u = \varnothing$, then all $p+1$-minimal vertices (those in the lime box) are valid for Eve, and accelerated toward $(u^{(4),\text{even}})$ in ϕ^{even}; the iteration then resumes from u'''. Otherwise, vertices in $A_{u\uparrow} \backslash Y_u$ are lifted to the gray zone and end up being valid for Adam. In this case, the iteration continues from u'', which corresponds to the box just above u's yellow box.

Using the sketch above, one may establish the following result.

Theorem 11.4 (Simulation of Zielonka's algorithm)

Let $R \in \omega$ be the total number of recursive calls performed by Zielonka's algorithm. Starting from the minimal parity bi-progress measure in $(\mathcal{L}^{\text{lazy},n,\text{odd}}_{n^{d/2}}, \mathcal{L}^{\text{lazy},n,\text{even}}_{n^{d/2}})$, there exists a sequence of at most $n^2R + 2$ applications of local lifts, resets, and (valid) accelerations which leads to a bi-progress measure which has weakly converged.

We believe that our sketch is detailed enough so that a full proof can easily be extracted by the reader interested in doing so. We hope that it is as least convincing enough that the theorem indeed holds.

5 Conclusion and perspectives

In this chapter, we started by proving that the so-called *universal attractor decomposition* algorithm of Jurdziński and Morvan [JM20] can be simulated by running two value iterations in parallel, with no interaction between them. In particular, we obtained a new proof of its correctness, which, surprisingly, is based on *signatures* rather than attractor-decompositions[9].

Going further, we have investigated the question of simulating Zielonka's algorithm, which adds the *empty-set termination rule* (in the vocabulary of [JM20]) to considerably decrease the number of recursive calls. This led us to designing a generic accelerating operator for parity bi-progress measures, which follows naturally from their interleaved structure. We have then extracted from Zielonka's proof [Zie98] a different (*lazy*) construction of a monotonic graph, and showed that Zielonka's algorithm can be simulated by accelerated parallel value iterations over such structures.

The proofs for our two simulation results are very much artificial and their details are somewhat unpleasant. We believe however that they help motivate the study of (accelerated) value iteration algorithms based on parity bi-progress measures; we now give more details and further discussions.

A wide class of iterative algorithms. We consider the following operators over parity bi-progress measures. The names stand for "Update", "Reset", and "Accelerate".

- (U) These are the usual (backpropagating) operators from Chapter 1, applied to either coordinate of ϕ.
- (R) These set the image of some vertices to *smaller* positions. Formally, a pointwise minimum with some fixed progress measure is applied.
- (A) These correspond to the accelerations described above. Formally, these are a familly indexed by $(\mathrm{P}, p) \in \{\text{even}, \text{odd}\} \times [1, d+1]$, which perform the acceleration in $\phi^{\overline{\mathrm{P}}}$ over p-minimal vertices if they are all valid in ϕ^{P} (and act idly otherwise).

All operators above preserve the property of being smaller than both $\mathcal{L}^{\mathrm{P}}$-evaluations ψ^{P}.

For the sake of this discussion, we will use $\mathcal{A}^X$ when $X \subseteq \{U, R, A\}$ to refer to the class of iterative algorithms which allow the operators from X (and terminate when ϕ has weakly converged). Note that the underlying monotonic graphs $\mathcal{L}^{\text{odd}}$ and $\mathcal{L}^{\text{even}}$ are not fixed: different structures correspond to different algorithms and possible iterations (some of which may not be effective or even terminate; we stay at an informal level).

We have seen in Chapter 4 that the class $\mathcal{A}^{U,R}$ is actually quite easy to understand in terms of efficiency. (Note that this class contains the local lift operators which can be simulated by an update and a reset.) Indeed, thanks to the monotonicity of operators in U and R, the fastest iterations are simply those which iterate U; stated differently there is no point in resetting, optimal iterations in $\mathcal{A}^{U,R}$ belong to $\mathcal{A}^U$.

[9] We have not defined attractor-decompositions formally. In our language, an Eve-attractor-decomposition over a tree T^{odd} is a morphism $G_\sigma \to \mathcal{L}^{\text{lazy},n,\text{odd}}_{T^{\text{odd}}}$, where σ is a positional strategy (it is therefore winning). These are implicit in Section 4, but make no appearance in Section 2.

Note that usual (one-dimensional) iterations can be simulated in a weak fragment of $\mathcal{A}^{U,A}$ simply by updating for one player until reaching ψ^{P}, and then applying a single weak acceleration ($p = d+1$) to obtain weak convergence of the bi-progress measure. Therefore, the quasipolynomial value iteration algorithms from [CJK+17], [FJS+17], [JL17] or [Leh18; Par20] can be simulated in $\mathcal{A}^{U,A}$. Moreover, the main result of Section 2 states that the generic attractor-based algorithm of [JM20] can also be simulated in this fragment.

Section 4 establishes that Zielonka's algorithm can be simulated (with polynomial blow-up) in $\mathcal{A}^{U,A,R}$; we believe that this is also the case of the two other quasipolynomial attractor-based algorithms from [Par19] and [LSW19], but give no details to support this claim.

The issue of monotonicity. Therefore $\mathcal{A}^{U,R,A}$ contains both value iteration algorithms and attractor-based algorithms. A natural candidate for a fast iteration (in any pair of structures) is the *greedy* one: successively apply updates, and whenever possible, apply an acceleration. Stated differently, the pointwise maximum of all operators is applied at each iteration. It is easy to see that for any (reasonable) structure $(\mathcal{L}^{\mathrm{odd}}, \mathcal{L}^{\mathrm{even}})$ the acceleration operators are non-monotonous. Therefore it is no longer clear in general whether non-resetting iterations are faster than resetting ones when in the presence of accelerations.

There is a parallel here with strategy-improvement algorithms (see Chapter 9): there is no reason a priori that the greedy iteration is the fastest. In strategy improvements however it is well known in general that there is an iteration which converges in linearly-many steps. This is not clear in the current setting, but it is an interesting question: assuming knowledge of winning strategies, can we derive iterations in $\mathcal{A}^{U,R,A}$ which converge in polynomial time (assuming some fixed structure)?

We believe that there is hope in proving some (weak) monotonicity properties (for instance, monotonicity over progress measure which are accessible by an iteration) when restricting to some subclasses of structures. As of now, we conjecture this to be true for lazy monotonic graphs as defined in Section 4 but fail to pinpoint what aspect of the structure provides such a guarantee.

This would imply that over such a structure the greedy iteration is optimal or in other words that $\mathcal{A}^{U,R,A}$ collapses to $\mathcal{A}^{U,A}$ for optimal iterations. Since $\mathcal{A}^{U,R,A}$ captures many known algorithms for solving parity games – at least all known quasipolynomial algorithms, except maybe the recent priority promotion of Benerecetti, Dell'Erba, Mogavero, Schewe and Wojtczak [BDM+21] – understanding if such a result holds over some classes of monotonic graphs is well motivated.

Another possible direction would be trying to enrich our class of iterative algorithms with other accelerating operators. Indeed, it is not hard to come up with generalisations of Theorem 11.3 which allow for accelerations under various kinds of hypotheses; the structure of parity bi-progress measures is quite robust. Could we come up with a class of accelerating operators A' whose pointwise maximum is both efficiently computable and monotonic?

General conclusion

We have studied various aspects of (universal) monotonic graphs, in relation to turn-based games of infinite duration which are positionally determined for Eve. Monotonic graphs are linearly ordered graphs whose transitions are monotonous. Given a monotonic graph $\mathcal{L}$ over L and an arena G over V, one obtains a monotonous operator over the set L^V of progress measures, whose prefixpoints naturally yield uniform positional strategies for Eve.

If paths in $\mathcal{L}$ satisfy an objective W then obtained positional strategies are winning, and if $\mathcal{L}$ embeds all graphs satisfying W from a class of graphs $\mathcal{C}$ then arenas over $\mathcal{C}$ have such optimal strategies. This connects the study of games with (positional) objective W to the study of the W-universality question for monotonic graphs. Although the fixpoint-based approach is well-known, its formulation as a universality problem is recent (and has been fruitful for parity games), and the explicit introduction of monotonic graphs in this context is novel.

The first part culminates in a characterisation of positionality: a valuation (or objective) that admits a neutral letter is positional over arbitrary arenas if and only if it admits (non-uniform) universal monotonic graphs that are well-ordered. This is the first characterisation of (one-player) positionality. The positionality proof is based on the *strategy folding technique* which appears in the works of Emerson and Jutla [EJ91], Klarlund [Kla92] and later Walukiewiecz [Wal96] and Grädel and Walukiewicz [GW06]. The converse completeness result relies on (multiple) choice arenas which we introduced.

Besides illustrating this result by establishing (often known) positionality results for various valuations by means of universal monotonic graphs, we have shown how to lexicographically combine monotonic graphs so as to lift universality to the lexicographic product of the objectives. This yields a proof that positional objectives which admit a neutral letter are closed under lexicographic combinations; we are not aware of a direct proof for this result.

This motivates the quest for combinations of monotonic graphs which are well-behaved with respect to union (rather than lexicographical product). Such a general result would solve the main conjecture of Kopczyński [Kop06] (at least for positionality over arbitrary arenas). We believe that monotonic graphs can help make progress in this direction. We would also like to investigate whether universal monotonic graphs can be used to understand (one player) positionality over finite arenas. We refer the reader to the conclusion of Part I for more details about these two questions.

We then turned our attention to using (universal) monotonic graphs to devise algorithms for solving games with positional valuations (or objectives). A finite universal monotonic graph reduces solving games to computing a (least) fixpoint over the set of progress measures it induces.

The most direct way of doing so is by Kleene iteration, which is usually called value iteration (or progress measure lifting) in this context. These are the object of Part II, which establishes a connection between value iteration and Bojańczyk and Czerwiński's separating approach, and studies finite universal (monotonic) graphs which are minimal in size for various positionally determined conditions.

For parity games, we formalised the connection with universal trees, presented the construction of Jurdziński and Lazić [JL17] and the lower bound of Fijalkow [Fij18]. For mean-payoff games, we directly obtained the value iteration of Brim, Chaloupka, Doyen, Gentilini and Raskin [BCD+11], and also gave two other constructions and derived (almost) matching lower bounds. Our conclusion for mean-payoff games is that universal monotonic graphs are captured by the connection to energy games, in particular value iteration algorithms cannot lead to better algorithms.

Last, we investigated two cases where (specific) universal monotonic graphs can be combined by union. For mean-payoff parity games, we formalised the construction of Daviaud, Jurdziński and Lazić [DJL18] in our vocabulary. Besides leading to a universality proof template which we believe to be interesting in its own right, this allows for a completely asymmetric approach. For multi mean-payoff games in the (easier) lim sup semantic, we showed that a property of graphs (quantifier commutation) can be exploited to construct succinct universal graphs.

In Part III, which is composed of three independent chapters, we explored different possibilities, besides value iteration, for computing the sought fixpoint. In particular, parity and mean-payoff games are symmetric and not only positional but even bi-positional (over finite arenas), which we have not exploited so far.

We first discussed the strategy improvement paradigm. We prove that for valuations computed by monotonic graphs (fixpoint valuations), a simple application of the Knaster-Tarski theorem establishes that strategy improvement is applicable whenever the valuation is positionally determined for Adam over finite arenas (which is necessary). Such a simple characterisation appears to be novel; it allows to relax existence of a unique fixpoint (which underlies schemes based on reductions to discounted games) to a weaker condition. In particular, this shows that strategy improvements for mean-payoff games can be applied directly with the energy valuation, answering a question of Björklund anv Vorobyov [BV05] (which, perhaps surprisingly, appeared to be still open). This also motivates understanding which monotonic graphs correspond to co-positional (and thus bi-positional) valuations, in particular for deriving strategy improvements specific to parity games.

We then turned to mean-payoff games, advocating that the symmetry in mean-payoff games can be exploited by seeing potential transformations from the point of view of both players; stated differently, we simultaneously solve the energy game and its dual. Applying such an analysis, provided the arena is simple (it has no simple cycle with zero sum), we obtained a novel bound of $N+E^{+}+E^{-}+1$ on the number of iterations of the attractor-based algorithm of Gurvich, Karzanov and Khachiyan [GKK88] (GKK algorithm), which improves on the state of the art. We also showed that the technique of Dorfman, Kaplan and Zwick [DKZ19] for deriving the state-of-the-art combinatorial $O(m2^{n/2})$ runtime bound can be applied to the GKK algorithm.

We then gave a presentation of the strategy improvement algorithm of Schewe [Sch08] by means of energy games. This exploits – in a natural way, we believe – the fact that Dijkstra's algorithm, applicable when weights are non-negative, can be lifted to the two-player scenario without blow-up on its runtime. In particular, using retreat vertices (or escape arenas) as in [BV05] is not necessary. We call this variant of Schewe's algorithm the ESL algorithm (for Energy-Schewe-Luttenberger). This formulation naturally suggests an alternating primal-dual variant (AESL) of the same algorithm. Although preliminary experimental results are very encouraging, and all the more so for parity games, we are not able to establish its termination. We believe that further study of the GKK, ESL and AESL algorithms could be fruitful, and refer the reader to the conclusion of Chapter 10 for more discussion.

In the last chapter, we focused on attractor-based approaches for parity games, for which the symmetry is more intricate (interleaved, to be precise) than for mean-payoff games. We showed that the universal attractor-decomposition algorithm of Jurdziński and Morvan [JM20] can be simulated by (non-interacting) parallel value iterations, establishing in particular its correctness without formally appealing (even implicitly) to attractor-decompositions. Pursuing further the connection between attractor-based and value iteration algorithms, we examined the structure of parity bi-progress measure (comprised of one progress measure for each player), and devised a general way of using information from each iteration to accelerate the other one, whatever the underlying monotonic graphs (satisfying the parity condition and its complement).

This allowed us to simulate in a similar way Zielonka's algorithm by means of accelerated bi-progress measures. Many algorithms (including almost all quasipolynomial algorithms to date), but also novel non-discarding attractor-based algorithms, can thus be simulated by accelerated bi-progress measures. The study of this class of algorithms appears to be interesting and exciting; further discussion can be found in the conclusion of Chapter 11.

All in all, we advocate for the systematic study of monotonic graphs, their properties, and how to combine them. Besides well-ordered monotonic graphs which are relevant to positionality, we have overall fell short of defining interesting subclasses of monotonic graphs. We hold responsible a lack of time rather than inherent difficulty of such an endeavour, which we see as both exciting and manageable; we hope that further developments in the (close) future will support this claim.

Unrelated work

During my three years of PhD, I have had the occasion and the pleasure of working on a variety of unrelated subjects. I chose to focus my dissertation only on infinite duration games, for which I feel my understanding and personal contribution to be by far the largest. However this is not reflected in my list of publications (see below), and a large part of the research which is presented in the thesis is not (yet) published.

Below a brief discussion of subjects on which I have worked during my doctorate and which are not accounted for in this manuscript.

- *Invariants for linear loops.* My work in this field started during my 12-week internship in Oxford in 2016, under the supervision of Joël Ouaknine and James Worrell, and together with Nathanaël Fijalkow and Amaury Pouly. Back then, we characterised the existence of complex semialgebraic invariants which led to [FOO+17; FOO+19]. More recently, and joining forces with Engel Lefaucheux, we solved the much more challenging case of complex semilinear invariants [FLO+19] (a full version is currently under review).

- *Hankel matrices and lower bounds for arithmetic circuits.* This work was initiated during my master's internship in 2018 under the supervision of my two PhD advisors Olivier Serre and Nathanaël Fijalkow, and together with Guillaume Lagarde. The starting point was to formalise (see [FLO18] for details) similarities between the celebrated results of Fliess [Fli74] and Nisan [Nis91], respectively in the contexts of weighted word automata and of non-commutative algebraic branching programs. This correspondance was then lifted between weighted tree automata on one hand and non-associative arithmetic circuits on the other, where the interpretation of a more general result of Bozapalidis and Louscou-Bozapalidou [BL83] led to a novel characterization of the size of non-associative circuits. Exploiting this characterization we obtained strong new lower bounds both for non-commutative and (set-multilinear) commutative arithmetic circuits [FLO+20].

- *Stochastic population control.* In this joint work with Thomas Colcombet and Nathanaël Fijalkow [CFO20], we proved the decidability of a control problem, introduced and left open by Bertrand, Dewaskar, Genest, Gimbert and Godbole [BDG+19], in the context of populations of Markov decision processes. Along the way we introduced the sequential flow problem (see also full version available in [CFO19] and currently under review) which we believe to be of independent interest and whose complexity, despite some efforts, has yet to be settled.

- *Search algorithm for program synthesis.* Together with Nathanaël Fijalkow and Guillaume Lagarde, we propose a new deterministic algortihm as well as an improved sampling algorithm for exploring the space of programs in the context of their automatic synthesis from input/output. A conference submission is under review.

Personal references

[AFGL+21] A. Anand, N. Fijalkow, A. Goubault-Larrecq, J. Leroux, and P. Ohlmann. "New Algorithms for Combinations of Objectives using Separating Automata". In: *GandALF*. 2021 (cited on pp. 112, 148, 155).

[CFG+21] T. Colcombet, N. Fijalkow, P. Gawrychowski, and P. Ohlmann. "The Theory of Universal Graphs for Infinite Duration Games". In: *CoRR* abs/2104.05262 (2021) (cited on pp. 25, 112, 117).

[CFO19] T. Colcombet, N. Fijalkow, and P. Ohlmann. "Controlling a random population". In: *CoRR* abs/1911.01195 (2019) (cited on p. 231).

[CFO20] T. Colcombet, N. Fijalkow, and P. Ohlmann. "Controlling a Random Population". In: *FOSSACS*. Vol. 12077. Lecture Notes in Computer Science. Springer, 2020, pp. 119–135 (cited on p. 231).

[FGO20] N. Fijalkow, P. Gawrychowski, and P. Ohlmann. "Value Iteration Using Universal Graphs and the Complexity of Mean Payoff Games". In: *MFCS*. Vol. 170. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020, 34:1–34:15 (cited on p. 112).

[FLO+19] N. Fijalkow, E. Lefaucheux, P. Ohlmann, J. Ouaknine, A. Pouly, and J. Worrell. "On the Monniaux Problem in Abstract Interpretation". In: *SAS*. Vol. 11822. Lecture Notes in Computer Science. Springer, 2019, pp. 162–180 (cited on p. 231).

[FLO+20] N. Fijalkow, G. Lagarde, P. Ohlmann, and O. Serre. "Lower Bounds for Arithmetic Circuits via the Hankel Matrix". In: *STACS*. Vol. 154. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020, 24:1–24:17 (cited on p. 231).

[FLO18] N. Fijalkow, G. Lagarde, and P. Ohlmann. "Tight Bounds using Hankel Matrix for Arithmetic Circuits with Unique Parse Trees". In: *Electron. Colloquium Comput. Complex.* 25 (2018), p. 38 (cited on p. 231).

[FOO+17] N. Fijalkow, P. Ohlmann, J. Ouaknine, A. Pouly, and J. Worrell. "Semialgebraic Invariant Synthesis for the Kannan-Lipton Orbit Problem". In: *STACS*. Vol. 66. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017, 29:1–29:13 (cited on p. 231).

[FOO+19] N. Fijalkow, P. Ohlmann, J. Ouaknine, A. Pouly, and J. Worrell. "Complete Semialgebraic Invariant Synthesis for the Kannan-Lipton Orbit Problem". In: *Theory Comput. Syst.* 63.5 (2019), pp. 1027–1048 (cited on p. 231).

[JMO+20] M. Jurdziński, R. Morvan, P. Ohlmann, and K. S. Thejaswini. "A symmetric attractor-decomposition lifting algorithm for parity games". In: *CoRR* abs/2010.08288 (2020) (cited on p. 202).

Bibliography

[AAH+14] P. A. Abdulla, M. F. Atig, P. Hofman, R. Mayr, K. N. Kumar, and P. Totzke. "Infinite-state energy games". In: *CSL-LICS*. ACM, 2014, 7:1–7:10 (cited on p. 24).

[ABG+13] X. Allamigeon, P. Benchimol, S. Gaubert, and M. Joswig. *Tropicalizing the simplex algorithm*. 2013 (cited on p. 22).

[ABG14] X. Allamigeon, P. Benchimol, and S. Gaubert. "The Tropical Shadow-Vertex Algorithm Solves Mean Payoff Games in Polynomial Time on Average". In: *ICALP*. Vol. 8572. Lecture Notes in Computer Science. Springer, 2014, pp. 89–100 (cited on p. 22).

[AGK+18] X. Allamigeon, S. Gaubert, R. D. Katz, and M. Skomra. "Condition numbers of stochastic mean payoff games and what they say about nonarchimedean semidefinite programming". In: *CoRR* abs/1802.07712 (2018) (cited on p. 22).

[AGQ+21] M. Akian, S. Gaubert, Y. Qi, and O. Saadi. "Tropical linear regression and mean payoff games: or, how to measure the distance to equilibria". In: *CoRR* abs/2106.01930 (2021) (cited on p. 22).

[AGS18] X. Allamigeon, S. Gaubert, and M. Skomra. "Solving generic nonarchimedean semidefinite programs using stochastic game algorithms". In: *J. Symb. Comput.* 85 (2018), pp. 25–54 (cited on p. 22).

[AGS20] X. Allamigeon, S. Gaubert, and M. Skomra. "Tropical Spectrahedra". In: *Discret. Comput. Geom.* 63.3 (2020), pp. 507–548 (cited on p. 22).

[AKS87] I. Adler, R. M. Karp, and R. Shamir. "A simplex variant solving an m times d linear program in $O(min(m^2, d^2)$ expected number of pivot steps". In: *J. Complex.* 3.4 (1987), pp. 372–387 (cited on p. 22).

[AMS+13] P. A. Abdulla, R. Mayr, A. Sangnier, and J. Sproston. "Solving Parity Games on Integer Vectors". In: *CONCUR*. Vol. 8052. Lecture Notes in Computer Science. Springer, 2013, pp. 106–120 (cited on p. 24).

[AR17] B. Aminof and S. Rubin. "First-cycle games". In: *Inf. Comput.* 254 (2017), pp. 195–216 (cited on pp. 81, 108).

[AV06] D. Andersson and S. Vorobyov. "Fast Algorithms for Monotonic Discounted Linear Programs with Two Variables per Inequality". In: *Technical report*. Preprint NI06019-LAA. 2006 (cited on p. 20).

[BAMP81] M. Ben-Ari, Z. Manna, and A. Pnueli. "The Temporal Logic of Branching Time". In: POPL. Association for Computing Machinery, 1981, pp. 164–176 (cited on p. 12).

[Ban22] S. Banach. “Sur les opérations dans les ensembles abstraits et leur application aux équations intégrales”. In: *Fundamenta Mathematicae* 3 (1922), pp. 133–181 (cited on p. 80).

[BBR13] A. Bohy, V. Bruyère, and E. F.J.-F. Raskin. “Synthesis from LTL Specifications with Mean-Payoff Objectives”. In: *Tools and Algorithms for the Construction and Analysis of Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 169–184 (cited on p. 23).

[BBR21] B. Bordais, P. Bouyer, and S. L. Roux. “From local to global determinacy in concurrent graph games”. In: *CoRR* abs/2107.04081 (2021) (cited on p. 55).

[BC18] M. Bojańczyk and W. Czerwiński. *An Automata Toolbox*. `https://www.mimuw.edu.pl/~bojan/papers/toolbox-reduced-feb6.pdf`. 2018 (cited on pp. 14, 25, 111, 113, 116, 117).

[BCD+11] L. Brim, J. Chaloupka, L. Doyen, R. Gentilini, and J. Raskin. “Faster algorithms for mean-payoff games”. In: *Formal Methods in System Design* 38.2 (2011), pp. 97–118 (cited on pp. 21, 112, 117, 120, 139, 142, 170, 189, 194, 228).

[BCG+14] R. Bloem, K. Chatterjee, K. Greimel, T. A. Henzinger, G. Hofferek, B. Jobstmann, B. Könighofer, and R. Könighofer. “Synthesizing robust systems”. In: *Acta Informatica* 51.3-4 (2014), pp. 193–220 (cited on p. 23).

[BCH+09] R. Bloem, K. Chatterjee, T. A. Henzinger, and B. Jobstmann. “Better Quality in Synthesis through Quantitative Objectives”. In: *CAV*. Vol. 5643. Lecture Notes in Computer Science. Springer, 2009, pp. 140–156 (cited on p. 90).

[BCJ18] R. Bloem, K. Chatterjee, and B. Jobstmann. “Graph Games and Reactive Synthesis”. In: *Handbook of Model Checking*. Ed. by E. M. Clarke, T. A. Henzinge, H. Veith, and R. Bloem. Springer International Publishing, 2018, pp. 921–962 (cited on p. 11).

[BDG+19] N. Bertrand, M. Dewaskar, B. Genest, H. Gimbert, and A. A. Godbole. “Controlling a population”. In: *Logical Methods in Computer Science* 15.3 (2019) (cited on p. 231).

[BDM+09] P. Bouyer, M. Duflot, N. Markey, and G. Renault. “Measuring Permissivity in Finite Games”. In: *CONCUR*. Vol. 5710. Lecture Notes in Computer Science. Springer, 2009, pp. 196–210 (cited on p. 23).

[BDM+21] M. Benerecetti, D. Dell’Erba, F. Mogavero, S. Schewe, and D. Wojtczak. “Priority Promotion with Parysian Flair”. In: *CoRR* abs/2105.01738 (2021) (cited on p. 226).

[BDM16] M. Benerecetti, D. Dell’Erba, and F. Mogavero. “Solving Parity Games via Priority Promotion”. In: *CAV*. Vol. 9780. Lecture Notes in Computer Science. Springer, 2016, pp. 270–290 (cited on p. 15).

[BFL+08] P. Bouyer, U. Fahrenberg, K. G. Larsen, N. Markey, and J. Srba. “Infinite Runs in Weighted Timed Automata with Energy Constraints”. In: *FORMATS*. Ed. by F. Cassez and C. Jard. Vol. 5215. Lecture Notes in Computer Science. Springer, 2008, pp. 33–47 (cited on pp. 20, 21, 47, 79, 170).

[BFM+11] A. Bianco, M. Faella, F. Mogavero, and A. Murano. “Exploring the boundary of half-positionality”. In: *Ann. Math. Artif. Intell.* 62.1-2 (2011), pp. 55–77 (cited on p. 55).

[BHM+17] P. Bouyer, P. Hofman, N. Markey, M. Randour, and M. Zimmermann. “Bounding Average-Energy Games”. In: *FOSSACS*. Vol. 10203. Lecture Notes in Computer Science. 2017, pp. 179–195 (cited on p. 24).

[BHR+19] V. Bruyère, Q. Hautem, M. Randour, and J. Raskin. “Energy Mean-Payoff Games”. In: *CONCUR*. Vol. 140. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019, 21:1–21:17 (cited on p. 24).

[BJP+12] R. Bloem, B. Jobstmann, N. Piterman, A. Pnueli, and Y. Sa⊠ar. “Synthesis of Reactive(1) designs”. In: *Journal of Computer and System Sciences* 78.3 (2012), pp. 911–938 (cited on p. 12).

[BJW02] J. Bernet, D. Janin, and I. Walukiewicz. “Permissive strategies: from parity games to safety games”. In: *RAIRO Theor. Informatics Appl.* 36.3 (2002), pp. 261–275 (cited on pp. 117, 125).

[BL18] U. Boker and K. Lehtinen. “On the Way to Alternating Weak Automata”. In: *FSTTCS*. Vol. 122. LIPIcs. 2018, 21:1–21:22 (cited on p. 13).

[BL69] J. R. Büchi and L. Landweber. “Solving sequential conditions by finite-state strategies”. In: *Transactions of the American Mathematical Society* 138 (1969), pp. 295–311 (cited on pp. 10–12, 53, 54).

[BL83] S. Bozapalidis and O. Louscou-Bozapalidou. “The Rank of a Formal Tree Power Series”. In: *Theoretical Computer Science* 27 (1983), pp. 211–215 (cited on p. 231).

[Bla97] R. G. Bland. “New finite pivoting rules for the simpler method”. In: *Math. Oper. Res.* 2 (1997), pp. 103–107 (cited on p. 16).

[BLO+20] P. Bouyer, S. Le Roux, Y. Oualhadj, M. Randour, and P. Vandenhove. “Games Where You Can Play Optimally with Arena-Independent Finite Memory”. In: *CONCUR*. Vol. 171. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020, 24:1–24:22 (cited on p. 55).

[BLT21] P. Bouyer, S. Le Roux, and N. Thomasset. “Finite-memory strategies in two-player infinite games”. In: *CoRR* abs/2107.09945 (2021) (cited on p. 54).

[BMO+11] P. Bouyer, N. Markey, J. Olschewski, and M. Ummels. “Measuring Permissiveness in Parity Games: Mean-Payoff Parity Games Revisited”. In: *ATVA*. Vol. 6996. Lecture Notes in Computer Science. Springer, 2011, pp. 135–149 (cited on p. 23).

[BMR+15] P. Bouyer, N. Markey, M. Randour, K. G. Larsen, and S. Laursen. “Average-energy games”. In: *GandALF*. Vol. 193. EPTCS. 2015, pp. 1–15 (cited on p. 24).

[BMR14] V. Bruyère, N. Meunier, and J. Raskin. “Secure equilibria in weighted games”. In: *CSL-LICS*. ACM, 2014, 26:1–26:26 (cited on p. 90).

[BOR+21] P. Bouyer, Y. Oualhadj, M. Randour, and P. Vandenhove. “Arena-Independent Finite-Memory Determinacy in Stochastic Games”. In: *CONCUR*. Vol. 203. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021, 26:1–26:18 (cited on p. 55).

[BSV03] H. Björklund, S. Sandberg, and S. G. Vorobyov. “A Discrete Subexponential Algorithm for Parity Games”. In: *STACS*. Vol. 2607. Lecture Notes in Computer Science. Springer, 2003, pp. 663–674 (cited on p. 16).

[BSV04a] H. Björklund, S. Sandberg, and S. G. Vorobyov. “A Combinatorial Strongly Subexponential Strategy Improvement Algorithm for Mean Payoff Games”. In: *MFCS*. Vol. 3153. Lecture Notes in Computer Science. Springer, 2004, pp. 673–685 (cited on pp. 16, 20, 21, 170, 178).

[BSV04b] H. Björklund, S. Sandberg, and S. G. Vorobyov. “Memoryless determinacy of parity and mean payoff games: a simple proof”. In: *Theor. Comput. Sci.* 310.1-3 (2004), pp. 365–378 (cited on p. 19).

[BV01] E. Beffara and S. Vorobyov. *Is Randomized Gurvich-Karzanov-Khachiyan's Algorithm for Parity Games Polynomial?* Uppsala University, Sweden, 2001 (cited on pp. 174, 198).

[BV05] H. Björklund and S. G. Vorobyov. "Combinatorial structure and randomized subexponential algorithms for infinite games". In: *Theor. Comput. Sci.* 349.3 (2005), pp. 347–360 (cited on pp. 16, 189, 228).

[Büc77] J. R. Büchi. "Using Determinancy of Games to Eliminate Quantifiers". In: *FCT*. Vol. 56. Lecture Notes in Computer Science. Springer, 1977, pp. 367–378 (cited on pp. 10, 53).

[Büc83] J. R. Büchi. "State-Strategies for Games in F G". In: *J. Symb. Log.* 48.4 (1983), pp. 1171–1198 (cited on p. 89).

[CAH+03] A. Chakrabarti, L. de Alfaro, T. A. Henzinger, and M. Stoelinga. "Resource Interfaces". In: *EMSOFT*. Vol. 2855. Lecture Notes in Computer Science. Springer, 2003, pp. 117–133 (cited on pp. 20–22).

[Cas21] A. Casares. "On the Minimisation of Transition-Based Rabin Automata and the Chromatic Memory Requirements of Muller Conditions". In: *CoRR* abs/2105.12009 (2021) (cited on pp. 55, 96).

[CD10] K. Chatterjee and L. Doyen. "Energy Parity Games". In: *ICALP*. Vol. 6199. Lecture Notes in Computer Science. Springer, 2010, pp. 599–610 (cited on p. 23).

[CD12] K. Chatterjee and L. Doyen. "Energy parity games". In: *Theor. Comput. Sci.* 458 (2012), pp. 49–60 (cited on p. 23).

[CDF+18] W. Czerwiński, L. Daviaud, N. Fijalkow, M. Jurdziński, R. Lazić, and P. Parys. "Universal trees grow inside separating automata: Quasi-polynomial lower bounds for parity games". In: *CoRR* abs/1807.10546 (2018) (cited on pp. 14, 112, 117).

[CDH+10] K. Chatterjee, L. Doyen, T. A. Henzinger, and J. Raskin. "Generalized Mean-payoff and Energy Games". In: *FSTTCS*. Vol. 8. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2010, pp. 505–516 (cited on p. 23).

[CE81] E. Clarke and E. Emerson. "Design and synthesis of synchronisation skeletons using branching time temporal logic". In: *Lecture Notes in Computer Science* 131 (1981). Ed. by Springer, pp. 52–71 (cited on p. 12).

[CF18] T. Colcombet and N. Fijalkow. "Parity games and universal graphs". In: *CoRR* abs/1810.05106 (2018) (cited on pp. 25, 67, 117).

[CF19] T. Colcombet and N. Fijalkow. "Universal Graphs and Good for Games Automata: New Tools for Infinite Duration Games". In: *FoSSaCS*. 2019, pp. 1–26 (cited on pp. 25, 57).

[CFH14] T. Colcombet, N. Fijalkow, and F. Horn. "Playing Safe". In: *FSTTCS*. Vol. 29. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2014, pp. 379–390 (cited on pp. 54, 79, 85).

[CGG+05] A. Costan, S. Gaubert, E. Goubault, M. Martel, and S. Putot. "A Policy Iteration Algorithm for Computing Fixed Points in Static Analysis of Programs". In: *CAV*. Vol. 3576. Lecture Notes in Computer Science. Springer, 2005, pp. 462–475 (cited on p. 168).

[CGH+12] P. Cerný, S. Gopi, T. A. Henzinger, A. Radhakrishna, and N. Totla. "Synthesis from incompatible specifications". In: *EMSOFT*. ACM, 2012, pp. 53–62 (cited on p. 24).

[CH08] K. Chatterjee and T. A. Henzinger. "Value Iteration". In: *25 Years of Model Checking - History, Achievements, Perspectives*. Vol. 5000. Lecture Notes in Computer Science. Springer, 2008, pp. 107–138 (cited on p. 111).

[CHJ05] K. Chatterjee, T. A. Henzinger, and M. Jurdziński. "Mean-Payoff Parity Games". In: *LICS*. IEEE Computer Society, 2005, pp. 178–187 (cited on p. 23).

[CHP07] K. Chatterjee, T. A. Henzinger, and N. Piterman. "Generalized Parity Games". In: *FOSSACS*. Vol. 4423. Lecture Notes in Computer Science. Springer, 2007, pp. 153–167 (cited on p. 161).

[CHP08] K. Chatterjee, T. A. Henzinger, and N. Piterman. "Algorithms for Büchi Games". In: *CoRR* abs/0805.2620 (2008) (cited on p. 44).

[CHS17] K. Chatterjee, M. Henzinger, and A. Svozil. "Faster Algorithms for Mean-Payoff Parity Games". In: *MFCS*. Vol. 83. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017, 39:1–39:14 (cited on p. 23).

[Chu57] A. Church. "Application of Recursive Arithmetic to the Problem of Circuit Synthesis". In: *Summaries of the SISL* 1 (1957), pp. 3–50 (cited on p. 11).

[CJK+17] C. S. Calude, S. Jain, B. Khoussainov, W. Li, and F. Stephan. "Deciding parity games in quasipolynomial time". In: *STOC*. 2017, pp. 252–263 (cited on pp. 13, 14, 117, 135, 226).

[CJL+17] T. Colcombet, M. Jurdzinski, R. Lazic, and S. Schmitz. "Perfect half space games". In: *LICS*. IEEE Computer Society, 2017, pp. 1–11 (cited on pp. 22, 24, 90, 155).

[CN06] T. Colcombet and D. Niwiński. "On the positional determinacy of edge-labeled games". In: *Theor. Comput. Sci.* 352.1-3 (2006), pp. 190–196 (cited on p. 53).

[Con90] A. Condon. "On Algorithms for Simple Stochastic Games". In: *Advances In Computational Complexity Theory*. Vol. 13. DIMACS. DIMACS/AMS, 1990, pp. 51–71 (cited on p. 20).

[Con92] A. Condon. "The Complexity of Stochastic Games". In: *Inf. Comput.* 96.2 (1992), pp. 203–224 (cited on p. 16).

[CR15] C. Comin and R. Rizzi. "An Improved Pseudo-Polynomial Upper Bound for the Value Problem and Optimal Strategy Synthesis in Mean Payoff Games". In: *CoRR* abs/1503.04426 (2015) (cited on p. 21).

[CR16] C. Comin and R. Rizzi. "Faster $O(V^2EW)$-Time Energy Algorithms for Optimal Strategy Synthesis in Mean Payoff Games". In: *CoRR* abs/1609.01517 (2016) (cited on pp. 21, 198).

[CR17] C. Comin and R. Rizzi. "Improved Pseudo-polynomial Bound for the Value Problem and Optimal Strategy Synthesis in Mean Payoff Games". In: *Algorithmica* 77.4 (2017), pp. 995–1021 (cited on p. 21).

[CTGG99] J. Cochet-Terrasson, S. Gaubert, and J. Gunawardena. "A constructive fixed point theorem for min-max functions". In: *Dynamics and stability of systems* 14.4 (1999), pp. 407–433 (cited on p. 20).

[CV12] K. Chatterjee and Y. Velner. "Mean-Payoff Pushdown Games". In: *LICS*. IEEE Computer Society, 2012, pp. 195–204 (cited on pp. 24, 155).

[CV13] K. Chatterjee and Y. Velner. "Hyperplane Separation Technique for Multidimensional Mean-Payoff Games". In: *CONCUR*. Vol. 8052. Lecture Notes in Computer Science. Springer, 2013, pp. 500–515 (cited on p. 24).

[DG06] V. Dhingra and S. Gaubert. "How to solve large scale deterministic games with mean payoff by policy iteration". In: *VALUETOOLS*. Vol. 180. ACM International Conference Proceeding Series. ACM, 2006, p. 12 (cited on pp. 21, 22).

[Dij18a] T. van Dijk. "Attracting Tangles to Solve Parity Games". In: *CAV*. Vol. 10982. Lecture Notes in Computer Science. Springer, 2018, pp. 198–215 (cited on p. 15).

[Dij18b] T. van Dijk. "Oink: An Implementation and Evaluation of Modern Parity Game Solvers". In: *TACAS*. Vol. 10805. Lecture Notes in Computer Science. Springer, 2018, pp. 291–308 (cited on pp. 15, 194, 196, 197).

[DJL18] L. Daviaud, M. Jurdziński, and R. Lazić. "A pseudo-quasi-polynomial algorithm for mean-payoff parity games". In: *LICS*. 2018, pp. 325–334 (cited on pp. 23, 25, 112, 147, 148, 161, 228).

[DJL19] L. Daviaud, M. Jurdziński, and K. Lehtinen. "Alternating Weak Automata from Universal Trees". In: *CONCUR*. Vol. 140. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019, 18:1–18:14 (cited on pp. 14, 201, 202, 221).

[DJT20] L. Daviaud, M. Jurdziński, and K. S. Thejaswini. "The Strahler Number of a Parity Game". In: *ICALP*. Ed. by A. Czumaj, A. Dawar, and E. Merelli. Vol. 168. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020, 123:1–123:19 (cited on p. 171).

[DJW97] S. Dziembowski, M. Jurdzinski, and I. Walukiewicz. "How Much Memory is Needed to Win Infinite Games?" In: *LICS*. IEEE Computer Society, 1997, pp. 99–110 (cited on pp. 54, 55).

[DKZ19] D. Dorfman, H. Kaplan, and U. Zwick. "A Faster Deterministic Exponential Time Algorithm for Energy Games and Mean Payoff Games". In: *ICALP*. 2019, 114:1–114:14 (cited on pp. 21, 173, 174, 184, 185, 188, 198, 228).

[EH83] E. A. Emerson and J. Y. Halpern. ""Sometimes" and "Not Never" Revisited: On Branching versus Linear Time (Preliminary Report)". In: POPL. Association for Computing Machinery, 1983, pp. 127–140 (cited on p. 12).

[EJ88] E. A. Emerson and C. S. Jutla. "The Complexity of Tree Automata and Logics of Programs (Extended Abstract)". In: *FOCS*. IEEE Computer Society, 1988, pp. 328–337 (cited on p. 45).

[EJ89] E. Emerson and C. Jutla. "On simultaneously determinizing and complementing $omega$-automata". In: *Proceedings. Fourth Annual Symposium on Logic in Computer Science*. IEEE Computer Society, 1989, pp. 333–342 (cited on p. 10).

[EJ91] E. A. Emerson and C. S. Jutla. "Tree Automata, μ-Calculus and Determinacy". In: *FOCS*. IEEE Computer Society, 1991, pp. 368–377 (cited on pp. 10, 11, 13, 14, 53, 56, 63, 87, 89, 108, 125, 201, 227).

[EJS01] E. Emerson, C. S. Jutla, and A. Sistla. "On model checking for the µ-calculus and its fragments". In: *Theoretical Computer Science* 258.1 (2001), pp. 491–522 (cited on p. 11).

[EJS93] E. A. Emerson, C. S. Jutla, and A. P. Sistla. "On Model-Checking for Fragments of µ-Calculus". In: *Proceedings of the 5th International Conference on Computer Aided Verification*. CAV '93. Berlin, Heidelberg: Springer-Verlag, 1993, pp. 385–396 (cited on p. 11).

[EKS18] J. Esparza, J. Kretínský, and S. Sickert. "One Theorem to Rule Them All: A Unified Translation of LTL into ω-Automata". In: *LICS*. ACM, 2018, pp. 384–393 (cited on p. 12).

[EM73] A. Ehrenfeucht and J. Mycielski. "Positional games over a graph". In: *Notices of the American Mathematical Society* 20 (1973), A–334 (cited on pp. 19, 46, 53).

[EM79] A. Ehrenfeucht and J. Mycielski. "Positional strategies for mean payoff games". In: *International Journal of Game Theory* 109.8 (1979), pp. 109–113 (cited on pp. 19, 46, 81, 108).

[Eme85] E. A. Emerson. "Automata, tableaux, and temporal logics". In: *Logics of Programs*. Springer Berlin Heidelberg, 1985, pp. 79–88 (cited on p. 11).

[Fea10a] J. Fearnley. "Non-oblivious Strategy Improvement". In: *LPAR*. Vol. 6355. Lecture Notes in Computer Science. Springer, 2010, pp. 212–230 (cited on p. 17).

[Fea10b] J. Fearnley. "Strategy Iteration Algorithms for Games and Markov Decision Processes". PhD thesis. University of Warwick, 2010 (cited on p. 168).

[FHZ11] O. Friedmann, T. D. Hansen, and U. Zwick. "Subexponential lower bounds for randomized pivoting rules for the simplex algorithm". In: *STOC*. ACM, 2011, pp. 283–292 (cited on p. 17).

[Fij18] N. Fijalkow. "An Optimal Value Iteration Algorithm for Parity Games". In: *CoRR* abs/1801.09618 (2018) (cited on pp. 14, 25, 112, 125, 135, 228).

[Fin16] B. Finkbeiner. "Synthesis of Reactive Systems". In: *Dependable Software Systems Engineering*. 2016 (cited on p. 11).

[FJK+19] J. Fearnley, S. Jain, B. de Keijzer, S. Schewe, F. Stephan, and D. Wojtczak. "An ordered approach to solving parity games in quasi-polynomial time and quasi-linear space". In: *Int. J. Softw. Tools Technol. Transf.* 21.3 (2019), pp. 325–349 (cited on p. 13).

[FJL+11] U. Fahrenberg, L. Juhl, K. G. Larsen, and J. Srba. "Energy Games in Multiweighted Automata". In: *ICTAC*. Vol. 6916. Lecture Notes in Computer Science. Springer, 2011, pp. 95–115 (cited on p. 24).

[FJS+17] J. Fearnley, S. Jain, S. Schewe, F. Stephan, and D. Wojtczak. "An ordered approach to solving parity games in quasi polynomial time and quasi linear space". In: *SPIN*. 2017, pp. 112–121 (cited on pp. 13, 14, 117, 135, 226).

[FL09] O. Friedmann and M. Lange. "Solving Parity Games in Practice". In: *ATVA*. Vol. 5799. Lecture Notes in Computer Science. Springer, 2009, pp. 182–196 (cited on pp. 15, 16).

[Fli74] M. Fliess. "Matrices de Hankel". In: *Journal de Mathématiques Pures et Appliquées* 53 (1974), pp. 197–222 (cited on p. 231).

[Fri09] O. Friedmann. "An Exponential Lower Bound for the Parity Game Strategy Improvement Algorithm as We Know it". In: *LICS*. IEEE Computer Society, 2009, pp. 145–156 (cited on pp. 17, 170, 199).

[Fri11a] O. Friedmann. "A Subexponential Lower Bound for Zadeh's Pivoting Rule for Solving Linear Programs and Games". In: *ICPO*. Vol. 6655. Lecture Notes in Computer Science. Springer, 2011, pp. 192–206 (cited on p. 17).

[Fri11b] O. Friedmann. "Exponential Lower Bounds for Solving Infinitary Payoff Games and Linear Programs". PhD thesis. Ludwig Maximilians University Munich, 2011 (cited on pp. 17, 168, 169).

[Fri13] O. Friedmann. “A superpolynomial lower bound for strategy iteration based on snare memorization”. In: *Discret. Appl. Math.* 161.10-11 (2013), pp. 1317–1337 (cited on p. 17).

[FT84] M. L. Fredman and R. E. Tarjan. “Fibonacci Heaps and Their Uses in Improved Network Optimization Algorithms”. In: *FOCS*. IEEE Computer Society, 1984, pp. 338–346 (cited on p. 193).

[FZ14] N. Fijalkow and M. Zimmermann. “Parity and Streett Games with Costs”. In: *Log. Methods Comput. Sci.* 10.2 (2014) (cited on p. 23).

[Gal58] T. Gallai. “Maximum-Minimum Sätze über Graphen”. In: *Acta Math. Acad. Sci. Hung.* 9 (1958), pp. 395–434 (cited on p. 21).

[GG98] S. Gaubert and J. Gunawardena. “The duality theorem for min-max functions”. In: *C. R. Acad. Sci. Paris Sér. I Math.* 1 (1998), pp. 43–48 (cited on p. 20).

[GH82] Y. Gurevich and L. Harrington. “Trees, Automata, and Games”. In: STOC. Association for Computing Machinery, 1982, pp. 60–65 (cited on pp. 10, 14, 45, 53, 54).

[Gil57] D. Gilette. “Stochastic games with zero stop probabilities”. In: *Contributions to the Theory of Games* 3 (1957), pp. 179–187 (cited on p. 19).

[Gim07] H. Gimbert. “Jeux positionnels”. PhD thesis. Université Paris Diderot, 2007 (cited on p. 53).

[GKK88] V. A. Gurvich, A. V. Karzanov, and L. G. Khachiyan. “Cyclic games and an algorithm to find minimax cycle means in directed graphs”. In: *USSR Computational Mathematics and Mathematical Physics* 28 (1988), pp. 85–91 (cited on pp. 19, 21, 26, 81, 173, 175, 181, 184–186, 198, 228).

[GLM+15] J. Gajarský, M. Lampis, K. Makino, V. Mitsou, and S. Ordyniak. “Parameterized Algorithms for Parity Games”. In: *MFCS*. Vol. 9235. Lecture Notes in Computer Science. Springer, 2015, pp. 336–347 (cited on p. 15).

[GS53] D. Gale and F. Stewart. “Infinite games with perfect information”. In: *Contributions to the Theory of Games*. Vol. 28. 2. Princeton University Press., 1953, pp. 245–266 (cited on p. 40).

[GTW02] E. Grädel, W. Thomas, and T. Wilke, eds. *Automata, Logics, and Infinite Games: A Guide to Current Research [outcome of a Dagstuhl seminar, February 2001]*. Vol. 2500. Lecture Notes in Computer Science. Springer, 2002 (cited on p. 10).

[GW06] E. Grädel and I. Walukiewicz. “Positional Determinacy of Games with Infinitely Many Priorities”. In: *Log. Methods Comput. Sci.* 2.4 (2006) (cited on pp. 42, 53, 56, 89, 227).

[GW13] M. Gazda and T. A. C. Willemse. “Zielonka’s Recursive Algorithm: dull, weak and solitaire games and tighter bounds”. In: *GandALF*. Vol. 119. EPTCS. 2013, pp. 7–20 (cited on p. 14).

[GZ05] H. Gimbert and W. Zielonka. “Games Where You Can Play Optimally Without Any Memory”. In: *CONCUR*. Vol. 3653. Lecture Notes in Computer Science. Springer, 2005, pp. 428–442 (cited on pp. 53–55, 79, 81, 171).

[HD05] P. Hunter and A. Dawar. “Complexity Bounds for Regular Games”. In: *MFCS*. Vol. 3618. Lecture Notes in Computer Science. Springer, 2005, pp. 495–506 (cited on p. 45).

[HK66] A. J. Hoffman and R. M. Karp. “On Nonterminating Stochastic Games”. In: *Management Science* 12.5 (1966), pp. 359–370 (cited on p. 16).

[HM80] M. Hennessy and R. Milner. “On observing nondeterminism and concurrency”. In: *Automata, Languages and Programming*. Springer Berlin Heidelberg, 1980, pp. 299–309 (cited on p. 10).

[HMZ13] T. D. Hansen, P. B. Miltersen, and U. Zwick. “Strategy Iteration Is Strongly Polynomial for 2-Player Turn-Based Stochastic Games with a Constant Discount Factor”. In: *J. ACM* 60.1 (2013), 1:1–1:16 (cited on pp. 16, 49, 169, 175).

[How60] R. Howard. *Dynamic programming and Markov processes*. Technology Press of Massachusetts Institute of Technology, 1960 (cited on p. 16).

[JL17] M. Jurdziński and R. Lazić. “Succinct progress measures for solving parity games”. In: *LICS*. 2017, pp. 1–9 (cited on pp. 13–15, 25, 112, 117, 125, 130, 135, 148, 158, 171, 201, 226, 228).

[JLR13] L. Juhl, K. G. Larsen, and J. Raskin. “Optimal Bounds for Multiweighted and Parametrised Energy Games”. In: *Theories of Programming and Formal Methods - Essays Dedicated to Jifeng He on the Occasion of His 70th Birthday*. Vol. 8051. Lecture Notes in Computer Science. Springer, 2013, pp. 244–255 (cited on p. 24).

[JLS15] M. Jurdzinski, R. Lazic, and S. Schmitz. “Fixed-Dimensional Energy Games are in Pseudo-Polynomial Time”. In: *ICALP*. Vol. 9135. Lecture Notes in Computer Science. Springer, 2015, pp. 260–272 (cited on p. 24).

[JM20] M. Jurdziński and R. Morvan. “A Universal Attractor Decomposition Algorithm for Parity Games”. In: *CoRR* abs/2001.04333 (2020) (cited on pp. 15, 26, 201–205, 207, 209, 210, 216, 225, 226, 229).

[JPZ06] M. Jurdzinski, M. Paterson, and U. Zwick. “A deterministic subexponential algorithm for solving parity games”. In: *SODA*. ACM Press, 2006, pp. 117–123 (cited on pp. 13, 15).

[Jur00] M. Jurdziński. “Small Progress Measures for Solving Parity Games”. In: *STACS*. 2000, pp. 290–301 (cited on pp. 13, 14, 21, 120, 125, 170).

[Jur98] M. Jurdziński. “Deciding the winner in parity games is in UP $\cap$ co-UP”. In: *Information Processing Letters* 68.3 (1998), pp. 119–124 (cited on pp. 11, 22).

[Kal92] G. Kalai. “A Subexponential Randomized Simplex Algorithm (Extended Abstract)”. In: *STOC*. 1992, pp. 475–482 (cited on p. 16).

[KGZ06] L. Khachiyan, V. Gurvich, and J. Zhao. “Extending Dijkstra’s Algorithm to Maximize the Shortest Path by Node-Wise Limited Arc Interdiction”. In: *CSR*. Vol. 3967. Lecture Notes in Computer Science. Springer, 2006, pp. 221–234 (cited on p. 192).

[KK91] N. Klarlund and D. Kozen. “Rabin Measures and Their Applications to Fairness and Automata Theory”. In: *LICS*. IEEE Computer Society, 1991, pp. 256–265 (cited on pp. 56, 87, 108).

[KL21] Z. K. Koh and G. Loho. *Beyond Value Iteration for Parity Games: Strategy Iteration with Universal Trees*. 2021 (cited on p. 171).

[KL93] A. Karzanov and V. Lebedev. “Cyclical games with prohibitions”. In: *Math. Programming* 60 (1993), pp. 277–293 (cited on p. 19).

[Kla91] N. Klarlund. “Progress Measures for Complementation of omega-Automata with Applications to Temporal Logic”. In: *FOCS*. IEEE Computer Society, 1991, pp. 358–367 (cited on pp. 56, 87, 201, 202, 221).

[Kla92] N. Klarlund. “Progress Measures, Immediate Determinacy, and a Subset Construction for Tree Automata”. In: *LICS*. IEEE Computer Society, 1992, pp. 382–393 (cited on pp. 53, 56, 64, 87, 227).

[Kop06] E. Kopczyński. “Half-Positional Determinacy of Infinite Games”. In: *ICALP*. 2006, pp. 336–347 (cited on pp. 53–56, 81, 89, 96, 147, 155, 227).

[Koz21a] A. Kozachinskiy. “Continuous Positional Payoffs”. In: *CONCUR*. Vol. 203. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021, 10:1–10:17 (cited on pp. 54, 85, 108, 169, 178).

[Koz21b] A. Kozachinskiy. “Polyhedral Value Iteration for Discounted Games and Energy Games”. In: *SODA*. SIAM, 2021, pp. 600–616 (cited on pp. 21, 174, 185).

[Koz83] D. Kozen. “Results on the propositional μ-calculus”. In: *Theoretical Computer Science* 27.3 (1983), pp. 333–354 (cited on pp. 10, 12).

[KT28] B. Knaster and A. Tarski. “Un théorème sur les fonctions d’ensembles”. In: *Ann. Soc. Polon. Math.* 6 (1928), pp. 133–134 (cited on p. 31).

[Kup12] O. Kupferman. “Recent Challenges and Ideas in Temporal Synthesis”. In: *SOFSEM*. 2012 (cited on p. 12).

[KV05] O. Kupferman and M. Vardi. “Safraless Decision Procedures”. In: vol. 2005. FOCS. Nov. 2005, pp. 531–540 (cited on p. 12).

[Kön27] D. König. “Über eine Schlussweise aus dem Endlichen ins Unendliche”. In: *Acta Sci. Math.* (1927) (cited on p. 35).

[LB20] K. Lehtinen and U. Boker. “Register Games”. In: *Logical Methods in Computer Science* 16.2 (2020) (cited on p. 13).

[LDT14] Y. Liu, Z. Duan, and C. Tian. “An Improved Recursive Algorithm for Parity Games”. In: *TASE*. IEEE Computer Society, 2014, pp. 154–161 (cited on pp. 220, 223).

[Leb16] V. Lebedev. “Exponential examples of solving parity games”. In: *Computational Mathematics and Mathematical Physics* 56 (2016), pp. 688–697 (cited on p. 174).

[Leh18] K. Lehtinen. “A modal-μ perspective on solving parity games in quasi-polynomial time”. In: *LICS*. 2018, pp. 639–648 (cited on pp. 13, 117, 226).

[LL69] T. M. Liggett and S. A. Lippman. “Stochastic Games with Perfect Information and Time Average Payoff”. In: *SIAM Review* 11.4 (1969), pp. 604–607 (cited on p. 19).

[LMS20] M. Luttenberger, P. J. Meyer, and S. Sickert. “Practical synthesis of reactive systems from LTL specifications via parity games”. In: *Acta Informatica* 57.1-2 (2020), pp. 3–36 (cited on pp. 12, 17, 26, 189, 199).

[Loh17] G. Loho. “Combinatorics of tropical linear programming”. PhD thesis. Technische Universität Berlin, 2017 (cited on p. 22).

[LP07] Y. Lifshits and D. Pavlov. “Potential theory for mean payoff games”. In: *Journal of Mathematical Sciences* 145 (2007), pp. 4967–4974 (cited on pp. 21, 173).

[LPS+21] K. Lehtinen, P. Parys, S. Schewe, and D. Wojtczak. “A Recursive Approach to Solving Parity Games in Quasipolynomial Time”. In: *CoRR* abs/2104.09717 (2021) (cited on pp. 16, 201).

[LSW19] K. Lehtinen, S. Schewe, and D. Wojtczak. "Improving the complexity of Parys' recursive algorithm". In: *CoRR* abs/1904.11810 (2019) (cited on pp. 15, 201, 226).

[Lud95] W. Ludwig. "A Subexponential Randomized Algorithm for the Simple Stochastic Game Problem". In: *Inf. Comput.* 117.1 (1995), pp. 151–155 (cited on pp. 16, 20).

[Lut08] M. Luttenberger. "Strategy Iteration using Non-Deterministic Strategies for Solving Parity Games". In: *CoRR* abs/0806.2923 (2008) (cited on pp. 12, 16, 22, 26, 170, 174, 189, 192).

[Mar75] D. A. Martin. "Borel Determinacy". In: *Annals of Mathematics* 102.2 (1975), pp. 363–371 (cited on p. 40).

[McN67] R. McNaughton. "Project MAC Progress Report V, Massachusetts Institute of Technology". In: (1967-1968) (cited on p. 11).

[McN93] R. McNaughton. "Infinite Games Played on Finite Graphs". In: *Annals of Pure and Applied Logic* 65.2 (1993), pp. 149–184 (cited on pp. 10, 14, 53).

[ML16] P. J. Meyer and M. Luttenberger. "Solving Mean-Payoff Games on the GPU". In: *ATVA*. Vol. 9938. Lecture Notes in Computer Science. 2016, pp. 262–267 (cited on pp. 21, 195, 196).

[Mos84] A. Mostowski. "Regular expressions for infinite trees and a standard form of automata". In: *Symposium on Computation Theory*. 1984 (cited on p. 11).

[Mos91] A. W. Mostowski. *Games with Forbidden Positions*. Tech. rep. 78. University of Gdansk, 1991 (cited on p. 11).

[Mou76] H. Moulin. "Prolongement des jeux à deux joueurs de somme nulle". PhD thesis. 1976 (cited on pp. 19, 81).

[MS87] D. E. Muller and P. E. Schupp. "Alternating automata on infinite trees". In: *Theoretical Computer Science* 54.2 (1987), pp. 267–276 (cited on p. 10).

[MSL18] P. J. Meyer, S. Sickert, and M. Luttenberger. "Strix: Explicit Reactive Synthesis Strikes Back!" In: *CAV*. Vol. 10981. Lecture Notes in Computer Science. Springer, 2018, pp. 578–586 (cited on pp. 12, 189, 199).

[MSW96] J. Matoušek, M. Sharir, and E. Welzl. "A Subexponential Bound for Linear Programming". In: *Algorithmica* 16.4/5 (1996), pp. 498–516 (cited on p. 16).

[Muc84] A. Muchnik. "Games on infinite trees and automata with dead-end markers—a new proof of the decidability of the monadic theory of two successors". In: *Semiotics and Information* 4 (1984), pp. 17–24 (cited on p. 10).

[Nis91] N. Nisan. "Lower Bounds for Non-Commutative Computation (Extended Abstract)". In: *Proceedings of the 23rd Symposium on Theory of Computing (STOC 1991)*. ACM, 1991, pp. 410–418 (cited on p. 231).

[Niw86] D. Niwinski. "On Fixed-Point Clones". In: *ICALP*. Vol. 226. Lecture Notes in Computer Science. Springer, 1986, pp. 464–473 (cited on p. 10).

[Niw88] D. Niwinski. "Fixed Points vs. Infinite Generation". In: *LICS*. IEEE Computer Society, 1988, pp. 402–409 (cited on p. 10).

[NYY92] A. Nerode, A. Yakhnis, and V. Yakhnis. "Concurrent Programs as Strategies in Games". In: *Logic from Computer Science*. Springer New York, 1992, pp. 405–479 (cited on p. 12).

[Obd03] J. Obdrzálek. "Fast Mu-Calculus Model Checking when Tree-Width Is Bounded". In: *CAV*. Vol. 2725. Lecture Notes in Computer Science. Springer, 2003, pp. 80–92 (cited on p. 15).

[Obd06] J. Obdrzálek. "DAG-width: connectivity measure for directed graphs". In: *SODA*. ACM Press, 2006, pp. 814–821 (cited on p. 15).

[Obd07] J. Obdrzálek. "Clique-Width and Parity Games". In: *CSL*. Vol. 4646. Lecture Notes in Computer Science. Springer, 2007, pp. 54–68 (cited on p. 15).

[Par19] P. Parys. "Parity Games: Zielonka's Algorithm in Quasi-Polynomial Time". In: *MFCS*. 2019, 10:1–10:13 (cited on pp. 15, 201, 226).

[Par20] P. Parys. "Parity Games: Another View on Lehtinen's Algorithm". In: *CSL*. 2020, 32:1–32:15 (cited on pp. 14, 117, 226).

[Pis99] N. N. Pisaruk. "Mean Cost Cyclical Games". In: *Mathematics of Operations Research* 24.4 (1999), pp. 817–828 (cited on pp. 20, 173).

[Pnu77] A. Pnueli. "The temporal logic of programs". In: FOCS 14 (1977), pp. 46–57 (cited on p. 11).

[PP04] D. Perrin and J. Pin. *Infinite words - automata, semigroups, logic and games*. Vol. 141. Pure and applied mathematics series. Elsevier Morgan Kaufmann, 2004 (cited on p. 43).

[PPS06] N. Piterman, A. Pnueli, and Y. Sa'ar. "Synthesis of Reactive(1) Designs". In: VMCAI. Springer-Verlag, 2006, pp. 364–380 (cited on p. 12).

[PR71] T. Parthasarathy and T. Raghavan. "Some Topics in Two-Person Games". In: (1971) (cited on p. 19).

[PR89] A. Pnueli and R. Rosner. "On the synthesis of a reactive module". In: POPL. 1989 (cited on p. 12).

[Pur95] A. Puri. "Theory of Hybrid Systems and Discrete Event Systems". PhD thesis. EECS Department, University of California, Berkeley, 1995 (cited on pp. 16, 19, 20, 81, 169).

[Rab69] M. O. Rabin. "Decidability of Second-Order Theories and Automata on Infinite Trees". In: *Transactions of the American Mathematical Society* 141 (1969), pp. 1–35 (cited on pp. 10, 11).

[Rab72] M. O. Rabin. *Automata on Infinite Objects and Church's Problem*. American Mathematical Society, 1972 (cited on p. 11).

[Ran14] M. Randour. "Synthesis in Multi-Criteria Quantitative Games". PhD thesis. Université de Mons, 2014 (cited on p. 22).

[RCN73] S. S. Rao, R. Chandrasekaran, and K. P. K. Nair. "Algorithms for discounted stochastic games". In: *Journal of Optimization Theory and Applications* 11.6 (1973), pp. 627–637 (cited on p. 16).

[RPR18] S. L. Roux, A. Pauly, and M. Randour. "Extending Finite-Memory Determinacy by Boolean Combination of Winning Conditions". In: *FSTTCS*. Vol. 122. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018, 38:1–38:20 (cited on p. 54).

[Saf88] S. Safra. "On the Complexity of Omega -Automata". In: *FOCS*. Vol. 29. IEEE Computer Society, 1988, pp. 319–327 (cited on p. 10).

[SB69] D. Scott and J. de Bakker. "A theory of programs". In: (1969). Unpublished manuscript (cited on p. 10).

[Sch07] S. Schewe. "Solving Parity Games in Big Steps". In: *FSTTCS*. Vol. 4855. Lecture Notes in Computer Science. Springer, 2007, pp. 449–460 (cited on p. 13).

[Sch08] S. Schewe. "An Optimal Strategy Improvement Algorithm for Solving Parity and Payoff Games". In: *CSL*. Vol. 5213. Lecture Notes in Computer Science. Springer, 2008, pp. 369–384 (cited on pp. 16, 20–22, 26, 170, 174, 189, 192, 194–196, 199, 228).

[SE84] R. S. Streett and E. A. Emerson. "The Propositional Mu-Calculus is Elementary". In: *Proceedings of the 11th Colloquium on Automata, Languages and Programming*. Springer-Verlag, 1984, pp. 465–472 (cited on p. 10).

[Ser04] O. Serre. "Contribution à l'étude des jeux sur des graphes de processus à pile". PhD thesis. Université Paris VIII Vincennes-Saint Denis, Nov. 2004 (cited on p. 41).

[SF07] S. Schewe and B. Finkbeiner. "Bounded Synthesis". In: ATVA. Springer-Verlag, 2007, pp. 474–488 (cited on p. 12).

[Sha53] L. S. Shapley. "Stochastic Games". In: 39.10 (1953), pp. 1095–1100 (cited on pp. 16, 19, 49, 53, 69, 80).

[Spe99] A. Specification. "Rev. 2.0". In: *ARM Limited* (1999) (cited on p. 12).

[STV15] S. Schewe, A. Trivedi, and T. Varghese. "Symmetric Strategy Improvement". In: *ICALP*. Vol. 9135. Lecture Notes in Computer Science. Springer, 2015, pp. 388–400 (cited on pp. 17, 26, 194).

[SWZ19] S. Schewe, A. Weinert, and M. Zimmermann. "Parity Games with Weights". In: *Log. Methods Comput. Sci.* 15.3 (2019) (cited on p. 23).

[Tar55] A. Tarski. "A lattice-theoretical fixpoint theorem and its applications". In: *Pacific Journal of Mathematics* 5.2 (1955), pp. 285–309 (cited on p. 31).

[Tho95] W. Thomas. "On the synthesis of strategies in infinite games". In: *STACS*. Springer Berlin Heidelberg, 1995, pp. 1–13 (cited on p. 12).

[Tho96] W. Thomas. *Languages, Automata and Logic*. 1996 (cited on p. 42).

[Var95] M. Y. Vardi. "An Automata-Theoretic Approach to Fair Realizability and Synthesis". In: *CAV*. Vol. 939. Lecture Notes in Computer Science. Springer, 1995, pp. 267–278 (cited on p. 12).

[VCD+15] Y. Velner, K. Chatterjee, L. Doyen, T. A. Henzinger, A. M. Rabinovich, and J. Raskin. "The complexity of multi-mean-payoff and multi-energy games". In: *Information and Computation* 241 (2015), pp. 177–196 (cited on pp. 25, 112, 155, 156, 159, 161).

[Vel15] Y. Velner. "Robust Multidimensional Mean-Payoff Games are Undecidable". In: *FoSSaCS*. Vol. 9034. Lecture Notes in Computer Science. Springer, 2015, pp. 312–327 (cited on p. 24).

[VJ00] J. Vöge and M. Jurdzinski. "A Discrete Strategy Improvement Algorithm for Solving Parity Games". In: vol. 1855. Lecture Notes in Computer Science. Springer, 2000, pp. 202–215 (cited on pp. 16, 17, 19, 20, 169, 170, 189).

[Vög00] J. Vöge. "Strategie synthese für Paritätsspiele auf endlichen Graphen". PhD thesis. University of Aachen, 2000 (cited on pp. 16, 20, 169, 170, 189).

[Wal96] I. Walukiewicz. “Pushdown Processes: Games and Model Checking”. In: *CAV*. Vol. 1102. Lecture Notes in Computer Science. Springer, 1996, pp. 62–74 (cited on pp. 13, 41, 56, 63, 89, 125, 170, 201, 227).

[Ye11] Y. Ye. “The Simplex and Policy-Iteration Methods Are Strongly Polynomial for the Markov Decision Problem with a Fixed Discount Rate”. In: *Math. Oper. Res.* 36.4 (2011), pp. 593–603 (cited on pp. 16, 49, 169).

[YY90] A. Yakhnis and V. Yakhnis. “Extension of Gurevich-Harrington’s Restricted Memory Determinacy Theorem: A Criterion for the Winning Player and an Explicit Class of Winning Strategies”. In: *Ann. Pure Appl. Log.* 48 (1990), pp. 277–297 (cited on p. 10).

[Zei94] S. Zeitman. “Unforgettable Forgetful Determinacy”. In: *Journal of Logic and Computation* 4.3 (June 1994), pp. 273–283 (cited on p. 10).

[Zie98] W. Zielonka. “Infinite Games on Finitely Coloured Graphs with Applications to Automata on Infinite Trees”. In: *Theoretical Computer Science* 200.1-2 (1998), pp. 135–183 (cited on pp. 14, 45, 53, 54, 56, 147, 194, 201, 221, 225).

[ZP95] U. Zwick and M. Paterson. “The Complexity of Mean Payoff Games”. In: *COCOON*. Vol. 959. Lecture Notes in Computer Science. Springer, 1995, pp. 1–10 (cited on pp. 16, 19, 20, 81).

[ZP96] U. Zwick and M. Paterson. “The Complexity of Mean Payoff Games on Graphs”. In: *Theoretical Computer Science* 158.1-2 (1996), pp. 343–359 (cited on pp. 16, 19, 20, 169).