Université de Mons
Faculté des Sciences
Département de Mathématique

F.R.S.-FNRS

Université Paris-Saclay
Laboratoire Méthodes Formelles
UMR 9021

Strategy Complexity of Zero-Sum Games on Graphs

Pierre VANDENHOVE

A dissertation submitted in fulfillment of the requirements of the degree of
Docteur en Sciences of the Université de Mons and
Docteur en informatique of the Université Paris-Saclay

April 2023

Jury

Christel BAIER   Reviewer
Technische Universität Dresden

Patricia BOUYER-DECITRE   Cosupervisor
CNRS, Laboratoire Méthodes Formelles

Véronique BRUYÈRE   President
UMONS – Université de Mons

Thomas COLCOMBET   Reviewer
CNRS, Institut de Recherche en Informatique Fondamentale

Laurent DOYEN   Examiner
CNRS, Laboratoire Méthodes Formelles

Benjamin MONMEGE   Examiner
Aix-Marseille Université, Laboratoire d'Informatique & Systèmes

Mickael RANDOUR   Cosupervisor
F.R.S.-FNRS, UMONS – Université de Mons

Abstract

We study two-player zero-sum turn-based games on graphs, a framework of choice in theoretical computer science. Such games model the possibly infinite interaction between a computer system (often called reactive) and its environment. The system, seen as a player, wants to guarantee a specification (translated to a game objective) based on the interaction; its environment is seen as an antagonistic opponent. The aim is to automatically synthesize a controller for the system that guarantees the specification no matter what happens in the environment, that is, a winning strategy in the derived game.

A crucial question in this synthesis quest is the complexity of strategies: when winning strategies exist for a game objective, how simple can they be, and how complex must they be? A standard measure of strategy complexity is the amount of memory needed to implement winning strategies for a given game objective. In other words, how much information should be remembered about the past to make optimal decisions about the future? Proving the existence of bounds on memory requirements has historically had a significant impact. Such bounds were, for instance, used to show the decidability of monadic second-order theories, and they are at the core of state-of-the-art synthesis algorithms. Particularly relevant are the finite-memory-determined objectives (for which winning strategies can be implemented with finite memory), as they allow for implementable controllers. In this thesis, we seek to further the understanding of finite-memory determinacy. We divide our contributions into two axes.

First, we introduce arena-independent finite-memory determinacy, describing the objectives for which a single automatic memory structure suffices to implement winning strategies in all games. We characterize this property through language-theoretic and algebraic properties of objectives in multiple contexts (games played on finite or infinite graphs). We show in particular that understanding the memory requirements in one-player game graphs (i.e., the simpler situation of games where the same player controls all the actions) usually leads to bounds on memory requirements in two-player zero-sum games. We also show that if we consider games played on infinite game graphs, the arena-independent-finite-memory-determined objectives are exactly the ω-regular objectives, providing a converse to the landmark result on finite-memory determinacy of ω-regular objectives. These results generalize previous works about the class of objectives requiring no memory to implement winning strategies.

Second, we identify natural classes of objectives for which precise memory requirements are surprisingly not fully understood. We introduce regular objectives (a subclass of the ω-regular objectives), which are simple objectives derived from regular languages. We effectively characterize their memory requirements for each player, and we study the computational complexity of deciding the existence of a small memory structure. We then move a step up in the complexity of the objectives and consider objectives definable with deterministic Büchi automata. We characterize the ones for which the first player needs no memory to implement winning strategies (a property called half-positionality). Thanks to this characterization, we show that half-positionality is decidable in polynomial time for this class of objectives. These results complement seminal results about memory requirements of classes of ω-regular objectives.

Keywords: game theory, controller synthesis, games on graphs, zero-sum games, optimal strategies, finite-memory strategies, finite-memory determinacy, ω-regular languages.

Résumé en français

Les jeux sur graphes à deux joueurs et à somme nulle constituent un modèle central en informatique théorique. De tels jeux modélisent une interaction potentiellement infinie entre un système dit réactif et son environnement. Le système est considéré comme un joueur et souhaite garantir une spécification (traduite en un objectif de jeu). Son environnement est considéré comme un joueur antagoniste. Le but est de synthétiser automatiquement un contrôleur pour le système qui garantit la spécification peu importe le comportement de l'environnement, ce qui correspond à construire une stratégie gagnante dans le jeu dérivé.

Une question cruciale dans cette problématique de synthèse est celle de la complexité des stratégies : si des stratégies gagnantes existent, à quel point peuvent-elles être simples et à quel point doivent-elles être complexes ? Une mesure standard de complexité des stratégies est la quantité de mémoire nécessaire pour implémenter des stratégies gagnantes pour un objectif donné. En d'autres termes, quelle quantité d'information faut-il retenir au sujet du passé pour prendre des décisions optimales concernant le futur ? Des preuves de l'existence de bornes sur les besoins en mémoire ont historiquement eu un impact important. Par exemple, de telles bornes ont mené à des preuves de décidabilité de théories monadiques du second ordre, et sont au cœur de nombreux algorithmes efficaces pour la synthèse. Les objectifs déterminés à mémoire finie (c'est-à-dire ceux qui admettent des stratégies gagnantes se limitant à une mémoire finie) sont particulièrement pertinents, car ils mènent à l'existence de contrôleurs pouvant être implémentés en pratique. Dans cette thèse, nous cherchons à améliorer la compréhension de la détermination à mémoire finie. Nous distinguons deux axes dans nos contributions.

Premièrement, nous introduisons le concept de détermination à mémoire finie indépendante de l'arène, qui décrit les objectifs pour lesquels une unique structure automatique de mémoire suffit pour implémenter des stratégies gagnantes dans tous les jeux. Nous caractérisons cette propriété via des propriétés algébriques et de langages dans différents contextes (jeux joués sur des graphes finis ou infinis). Nous montrons en particulier que la compréhension des besoins en mémoire dans les jeux à un joueur (c'est-à-dire les jeux plus simples dans lesquels le même joueur contrôle toutes les actions) mène généralement à des bornes sur les besoins en mémoire dans les jeux à deux joueurs et à somme nulle. Nous montrons également que si l'on considère les jeux joués sur des graphes infinis, les objectifs déterminés à mémoire finie indépendante de l'arène sont exactement les objectifs ω-réguliers, ce qui fournit une réciproque au célèbre théorème de détermination à mémoire finie de ces objectifs. Ces résultats généralisent des travaux précédents au sujet des objectifs pour lesquels aucune mémoire n'est nécessaire pour les stratégies gagnantes.

Deuxièmement, nous identifions des classes naturelles d'objectifs pour lesquels les besoins en mémoire ne sont pas complètement établis. Nous introduisons les objectifs réguliers (une sous-classe des ω-réguliers), qui sont des objectifs dérivés de langages réguliers. Nous donnons une caractérisation effective des besoins en mémoire de ces objectifs pour chacun des joueurs, et nous étudions la complexité de décider de l'existence d'une petite structure de mémoire. Nous considérons ensuite des objectifs plus complexes définissables avec des automates de Büchi déterministes. Nous caractérisons ceux pour lesquels le premier joueur n'a besoin d'aucune mémoire pour implémenter des stratégies gagnantes (une propriété appelée semi-positionnalité). Grâce à cette caractérisation,

nous montrons que la semi-positionnalité est décidable en temps polynomial pour ces objectifs. Ces résultats complètent des travaux fondateurs sur les besoins en mémoire des objectifs ω-réguliers.

Mots-clés : théorie des jeux, synthèse de contrôleurs, jeux sur graphes, jeux à somme nulle, stratégies optimales, stratégies à mémoire finie, langages ω-réguliers.

Acknowledgments

I am, first and foremost, deeply thankful to my supervisors, Patricia Bouyer and Mickael Randour. Working with just one of you would already have been fantastic, so I cannot overstate how lucky I am to work with the combination of you two. Thank you for your time, your guidance, and your advice.

I would like to sincerely thank all the jury members: Christel Baier, Véronique Bruyère, Thomas Colcombet, Laurent Doyen, and Benjamin Monmege. After spending a lot of time admiring your work, I am honored that you accepted to evaluate mine.

Even though only my name appears on the cover page, this work results in many ways from collaborations. Additionally to my supervisors, I am thankful to my stellar collaborators Antonio Casares, Nathanaël Fijalkow, Stéphane Le Roux, and Youssouf Oualhadj.

I am very fortunate for my great long-standing office coworkers Jérémy, Aline, Horacio, Clément, Gaëtan, James, Damien, and Benjamin as well as the newcomers Chloé, Nicolas, and Florent, without whom it would be tough to be as productive (in the long term). I also owe much to Thomas and Christian for their great advice all along my studies.

Finally, I thank my friends, with a special mention to Florent. I am grateful to my parents and my sister for always being there. This list would not be complete without thanking Alice for her infallible support.

Contents

Introduction 1

In this chapter, we introduce and motivate the model of games on graphs, its history, and its significance. We then present the question of strategy complexity in these games and relevant research questions, both solved and unsolved. We finally sketch the original contributions presented in this thesis and describe an outline of the thesis structure.

1.1 Context

Computer systems are often complex machines intertwining pieces of hardware and software. They are usually designed with specific purposes, and we rely on them to accomplish an increasingly large breadth of complex tasks. For some critical systems, failing to achieve what they were designed for may have drastic consequences, be they financial or lethal [BK08, Chapter 1]. Therefore, a crucial endeavor of computer science is to guarantee the *correct* behavior of computer systems, where correctness is with respect to some specified purpose.

[BK08]: Baier et al. (2008), *Principles of model checking*

Computer systems are often not isolated entities: they interact with their environment (whether through a network or through interactions with humans or other devices). They need to adapt and react to external events generated in this environment. Systems maintaining a continuous interaction with their surroundings are called *reactive* [HP84]. Reactivity is a common source of bugs and errors, as taking into account all possible events in the environment is notoriously difficult.

[HP84]: Harel et al. (1984), *On the Development of Reactive Systems*

Formal methods. There are multiple approaches seeking to increase confidence in computer systems. One of the most common is *testing*, which verifies that various scenarios lead to the expected behavior. However, there tend to be infinitely many different scenarios for reactive systems due to the number of combinations of events that the environment may produce. This means that testing reactive systems, while it can detect design mistakes, cannot guarantee the lack of them in general. The field of *formal methods* provides a more ambitious approach; its goal is to design

algorithms that give solid mathematical guarantees on the behavior of systems.

Verifying properties of computer systems is a hard problem, even undecidable in many cases of interest [Tur37]. A typical process using formal methods has a modeling phase: one first designs an abstraction of the computer system, simplifying the system while extracting as many of the relevant features for the property being checked. One of the successes of formal methods is to identify classes of expressive models that are amenable to being verified with respect to classes of expressive properties, as well as design efficient algorithms that do so.

[Tur37]: Turing (1937), *On Computable Numbers, with an Application to the Entscheidungsproblem*

Model checking and synthesis.   The traditional approach is then called *model checking*: given some model of a system and some *specification* of what the system should achieve, an algorithm is tasked with checking that the model indeed guarantees the specification no matter what happens in the environment (or, if stochasticity is used in the model, with a high enough probability). This approach assumes that the model represents a complete implementation of the system.

Another approach, which adds a layer of complexity, is *synthesis*. Synthesis starts from an *incomplete* implementation of a system, leaving some decisions to be made. To be complete, the system requires a *controller* that prescribes which actions to take in which situations. Given this incomplete description of the system, its environment, and a specification, the aim is to decide the existence of a controller for the system guaranteeing the specification and, if it does exist, automatically *synthesize* it. Correctness is then guaranteed by design of the controller. This is called the *(reactive) synthesis problem*. In this thesis, our goal is to contribute to the theoretical foundations of reactive synthesis.

Game theory.   A reactive system is an object with some capabilities; there are actions that it can perform influencing its future. Through these actions, it wants to fulfill a specification. Nevertheless, some of the events that may happen in the system's environment are beyond the system's control; they are seen as *uncontrollable*. Through these observations, we can model the interaction between the system and its environment as a *zero-sum game*. The computer system is a player, and a worst-case assumption is made about the environment, which is modeled as an antagonistic player. The specification is translated as a *game objective*. The system would like that the specification is guaranteed no matter what uncontrollable events happen in the environment. This corresponds to implementing a *winning strategy* for the objective in the derived zero-sum game opposing the system and the environment. The reactive synthesis problem can be reformulated as the quest to automatically construct winning strategies in the derived games, where the strategies correspond to controllers of the original system.

For the purpose of modeling possibly non-terminating systems, we assume that the interaction between the system and its environment lasts for an infinite duration. A reasonable way to encode a game objective is then to specify all infinite interactions between the system and the environment that are deemed acceptable with respect to the specification. This is usually done with *logical formulas* or *automata*.

The model. We consider *games on graphs*, which are central tools in theoretical computer science to model systems involving competing agents. More specifically, we study *two-player zero-sum games on graphs* played in a *turn-based* fashion.

In such a game, a graph represents the various states of a system and the transitions between them. A pebble in some vertex of the graph indicates the current state of the system. Two players, called $\mathcal{P}_1$ and $\mathcal{P}_2$, take turns moving the pebble along the edges of the graph. Exactly one player controls each vertex of the graph. Edges of the graph are labeled with *colors* from some given alphabet. We assume that the interaction between the players lasts for an infinite duration; as the pebble moves along the edges, it generates an infinite sequence of colors. Player $\mathcal{P}_1$ wants to achieve an *objective*, which is specified as a set of infinite sequences of colors deemed acceptable; the antagonistic player $\mathcal{P}_2$, as a worst-case assumption, seeks to prevent this.

We consider a simple and practical example. Assume that a computer system wants to let a user define a new password. To do so, the user must correctly enter the same valid password *twice in a row* after being prompted by the system. If and only if the user has correctly entered the same password twice, the system must immediately follow by providing a confirmation to the user. The system ($\mathcal{P}_1$) must therefore decide between *prompting* the user or *confirming* the inputs, based on the behavior of the user ($\mathcal{P}_2$) who may either *pass* or *fail* the password verification. We model this as a game on a graph. We assume that the alphabet of colors is {prompt, confirm, pass, fail}. The graph modeling the interaction between the players is depicted in Figure 1.1. The system controls the actions from vertex v_1 (where a choice can be made between prompt or confirm), and the user influences the action pass or fail from vertex v_2 (we leave the implementation of the actual function checking the passwords out of this model).
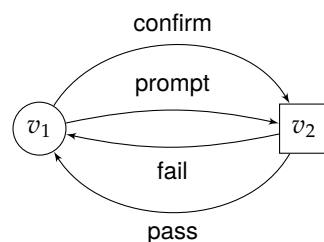
Figure 1.1: A two-player zero-sum game on a graph describing a system that must react accordingly when the same valid password is entered twice in a row.

The players use *strategies* to describe their behaviors, which are mathematical objects that make decisions based on the current vertex and what happened previously. We are here interested in building a winning strategy for the system, i.e., a strategy guaranteeing the specification no matter what the user does. This winning strategy needs to remember some information about the past, consisting in the previous actions of the players. One way to remember part of this information is by using a state machine updating its state depending on the events happening in the game. In this example, it is relevant for the state machine to count the number of times pass appeared in a row without fail, between 0 and 2. We depict such a machine in Figure 1.2.
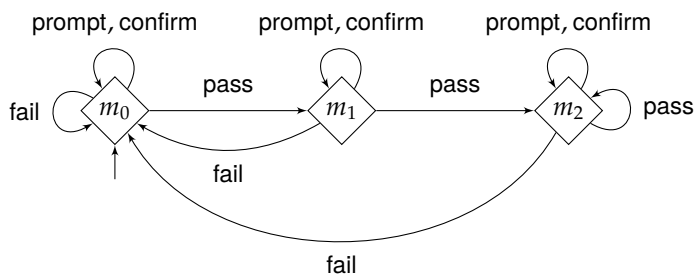
Figure 1.2: A state machine counting the number of consecutive occurrences of pass in order to output confirm at the adequate time.

This state machine has three states m_0, m_1, and m_2. It is initialized in state m_0. When an event occurs in the interaction, the machine state is updated by "reading" the event along a transition. For instance, if event pass occurs when the state is m_0, the state is updated to m_1. By knowing that the current state is m_1, the system knows that one pass has been seen with no fail since.

If the system does not remember anything about the past, it cannot make optimal decisions, as it needs to output sometimes prompt and sometimes confirm depending on the previous events in the environment. But using additionally the information provided by this machine, the system may make optimal decisions: when in m_0 or m_1, take action prompt; when in m_2, take action confirm.

A synthesis algorithm, given the graph and a formal encoding of the specification, should automatically output such a state machine and the decisions to make in each situation (i.e., for each pair composed of a graph vertex and a machine state). An interesting observation in this example is that it is possible to realize the specification with a machine with finitely many states. In other words, there is no need to remember all previous events in the interaction, and it suffices to remember the current state of a finite-state machine. This is a useful property to actually implement the strategy in a computer system.

1.2 Origin of games on graphs

We give an overview of the most fundamental works about the synthesis problem using games on graphs and its significance in practical applica-

tions. We refer to [GTW02; BCJ18] for a more detailed discussion about the significance of games on graphs.

Logical and automata-theoretic roots. The idea of the synthesis problem was born in the late 1950s, after Church [Chu57] wondered ambitiously about the existence of algorithms that could synthesize a logical circuit from mathematical specifications. The first answers came from works by Büchi [Büc62], Landweber [BL69], and Rabin [Rab69], who showed the decidability of multiple monadic second-order theories. The foundation at the core of these approaches is *automata theory*: specifications in these theories can be translated into automata [Büc60; Elg61; McN66]. These automata are often more complex than classical (non-)deterministic finite automata: they read *infinite* words that they have to accept or reject.

A reformulation of these questions through deciding the winner of *zero-sum turn-based games of infinite duration* arose as natural and intuitive, and offered simplifications of the above decidability proofs. A game-theoretic approach by Gurevich and Harrington [GH82] used games on infinite trees, which was also rephrased by McNaughton [McN93] using finite graphs. This thesis adopts the latter viewpoint due to its convenience and relative dominance in the modern literature.

Specifications from automata on infinite words are usually called *ω-regular*, as they extend the classical regular languages to words of infinite length. Multiple models of automata on infinite words (sometimes called *ω-automata*) have been defined to express *ω*-regular languages (*Büchi, Rabin, Streett, Muller, parity. . .*).

Reactive synthesis. These logical, automatic, and game-theoretic tools found practical applications to the fields of *model checking* and *synthesis* of *reactive systems* sketched above. Model checking [BK08] and synthesis [BCJ18] are extremely valuable approaches toward guaranteeing the correctness of systems. They offer automated proofs that (a model of an) implementation is correct; by analysis of the model for model checking, or by design for synthesis.

Deciding specifications using monadic second-order logic formulas unfortunately has non-elementary complexity [Mey75], making it impractical in such generality. The success of model checking and synthesis was helped by logical restrictions that were found to retain a good expressivity while being easier to decide. One successful example is the *linear-time temporal logic (LTL)* first defined by Pnueli [Pnu77], which is less expressive but for which the synthesis problem is "only" 2EXPTIME-complete [PR89]. Despite still a high theoretical complexity, recent endeavors (illustrated, e.g., by efficient algorithms for expressive fragments [BJP+12] and by the *Reactive Synthesis Competition SYNTCOMP*) show that the synthesis problem is feasible in practice for many reasonable LTL specifications.

[GTW02]: Grädel et al. (2002), *Automata, Logics, and Infinite Games: A Guide to Current Research [outcome of a Dagstuhl seminar, February 2001]*

[BCJ18]: Bloem et al. (2018), *Graph Games and Reactive Synthesis*

[Chu57]: Church (1957), *Application of Recursive Arithmetic to the Problem of Circuit Synthesis*

[Büc62]: Büchi (1962), *On a Decision Method in Restricted Second Order Arithmetic*

[BL69]: Büchi et al. (1969), *Definability in the Monadic Second-Order Theory of Successor*

[Rab69]: Rabin (1969), *Decidability of Second-Order Theories and Automata on Infinite Trees*

[Büc60]: Büchi (1960), *Weak Second-Order Arithmetic and Finite Automata*

[Elg61]: Elgot (1961), *Decision Problems of Finite Automata Design and Related Arithmetics*

[McN66]: McNaughton (1966), *Testing and Generating Infinite Sequences by a Finite Automaton*

[GH82]: Gurevich et al. (1982), *Trees, Automata, and Games*

[McN93]: McNaughton (1993), *Infinite Games Played on Finite Graphs*

[BK08]: Baier et al. (2008), *Principles of model checking*

[Mey75]: Meyer (1975), *Weak monadic second order theory of succesor is not elementary-recursive*

[Pnu77]: Pnueli (1977), *The Temporal Logic of Programs*

[PR89]: Pnueli et al. (1989), *On the Synthesis of a Reactive Module*

[BJP+12]: Bloem et al. (2012), *Synthesis of Reactive(1) designs*

In all these works, both about logical theories and about reactive synthesis, a key element for decidability can be understood as follows: when a player has a winning strategy in a game, this player has a winning strategy *that a finite-state machine can implement*. This usually provides an upper bound on the number of strategies to consider. The study of *how complex* strategies need to be in order to implement winning strategies (if they exist) becomes central.

1.3 Strategy complexity

Given a (two-player zero-sum) game on a graph, the first question one can ask is probably "which player can win?". More formally, this should be understood as "which player has a strategy that guarantees a win no matter how the opponent player behaves, i.e., a *winning strategy*?". Obviously, in a zero-sum game on a graph, the players cannot both have a winning strategy from the same vertex. A surprising property of most reasonable games is that they are *determined*, i.e., from any initial vertex, one of the players has a winning strategy (the negation would be that all strategies of both players are countered by a strategy of their opponent). This result stems from the *Borel determinacy* result by Martin [Mar75], stating that every turn-based *Borel game* is determined. A game is Borel if the game objective is a Borel set, which goes well beyond ω-regular languages.

[Mar75]: Martin (1975), *Borel determinacy*

With that out of the way, we know that in most games (including all the ones we will consider in this document), from a given initial vertex, exactly one of the players has a winning strategy. How do we decide which player it is? Answering this question is called *solving* the game. A reasonable attempt at an algorithm would be to try to exhibit a winning strategy for a player. However, strategies are complex objects. In general, a player's strategy needs to consider what already happened in the game (for instance, by looking at the sequence of edges already taken), and, when it is the player's turn, it needs to return the next edge to follow. There are therefore infinitely many strategies, and a winning strategy may not have a finite representation. To bound the size of the search space, we need stronger properties than just determinacy. This is where *finite-memory* determinacy proves useful.

Finite-memory determinacy. Finite-memory determinacy is the property of a game objective stating that, whenever a player can win in a graph for this objective, this player may also win using a "simple" strategy that uses only a *finite amount of memory*. In other words, it is not necessary to remember everything that happened in the past to win; a player can condense this unbounded information into a bounded amount of information and still retain sufficient information to make optimal decisions. In this case, we say that *finite memory suffices to win*. A common computational

model to implement finite-memory strategies is a *finite-state machine* (more precisely, a *Mealy machine* [Mea55]), which is roughly a finite automaton that also outputs the action to take depending on the current vertex of the graph and the current machine state.

A landmark result at the core of the decidability of the aforementioned logical theories is the *finite-memory determinacy of ω-regular game objectives*. The first authors to state a result in this form were Gurevich and Harrington [GH82], simplifying Rabin's proof [Rab69]. They showed that for games with objectives specified as *Muller conditions*, one of the players has a winning strategy implemented with a number of states at most factorial in the number of alphabet symbols on which the Muller condition is defined. Dziembowski, Jurdziński, and Walukiewicz have since refined this result [DJW97] to an algorithm that computes the precise number of memory states needed to implement winning strategies in Muller games (which is still factorial in the worst case).

Memoryless determinacy. Remarkably, for several canonical classes of games, even less memory is required: no information at all about the past is necessary to implement winning strategies (only the current vertex of the graph matters). Strategies that use no memory are called *memoryless* (or *positional*); when they suffice to win for an objective in all game graphs for both players, we say that the objective is *memoryless-determined*. A memoryless strategy is a special kind of a finite-memory strategy that uses only one memory state (and thus, cannot distinguish anything about the past). Arguably the most important games for which memoryless strategies suffice to win (for both players) are games with *parity conditions*. Parity conditions can be used to express all ω-regular languages when adjoined to an automaton (albeit with a larger automaton than when using Muller conditions), and games with parity conditions are effectively equivalent to the μ-calculus model checking [EJ91]. Their memoryless determinacy was shown independently by Emerson and Jutla [EJ91] and Mostowski [Mos91].

Memoryless determinacy of an objective is an attractive property in practice, as algorithms solving games with this objective do not need to consider complex strategies. A winning memoryless strategy is an object of size polynomial in the size of the graph: it suffices to store one action per vertex of the graph controlled by this player. As solving parity games where one player has fixed a strategy can be done in polynomial time, this leads to a direct proof that parity games are in NP ∩ coNP, just by analyzing the strategy complexity. Memoryless determinacy of parity games is partly why formulating specifications as deterministic parity automata is a common step of many efficient LTL synthesis algorithms [BCJ18]. Parity games have recently been shown to be solvable in quasi-polynomial time in a breakthrough result [CJK+17] — whether they are solvable in polynomial time remains an open problem.

[Mea55]: Mealy (1955), *A method for synthesizing sequential circuits*

[GH82]: Gurevich et al. (1982), *Trees, Automata, and Games*
[Rab69]: Rabin (1969), *Decidability of Second-Order Theories and Automata on Infinite Trees*

[DJW97]: Dziembowski et al. (1997), *How Much Memory is Needed to Win Infinite Games?*

[EJ91]: Emerson et al. (1991), *Tree Automata, Mu-Calculus and Determinacy (Extended Abstract)*
[Mos91]: Mostowski (1991), *Games with Forbidden Positions*

[BCJ18]: Bloem et al. (2018), *Graph Games and Reactive Synthesis*
[CJK+17]: Calude et al. (2017), *Deciding parity games in quasi-polynomial time*

Many other common classes of games are memoryless-determined when played over finite graphs; e.g., *discounted-sum* games [Sha53], *mean-payoff* games [EM79], *energy* games [CdHS03], *total-payoff* games [GZ04]. These games are used to model quantitative properties.

Given the simplicity of memoryless strategies, it is remarkable that they suffice for objectives as rich as the ones above. Following this observation, a lot of effort has been put in understanding which games admit memoryless optimal strategies, and in identifying the exact frontiers of *memoryless determinacy*. We mention, non-exhaustively, works by Gimbert and Zielonka [GZ05] (characterization of memoryless determinacy for games played on finite graphs), Colcombet and Niwiński [CN06] (characterization for games played on infinite graphs), Aminof and Rubin [AR17] (through the prism of *first-cycle games*), Kopczyński [Kop06], Bianco, Faella, Mogavero, and Murano [BFMM11], and Ohlmann [Ohl23] (*half-positionality*). All these advances were built by identifying the common underlying mechanisms in ad hoc proofs for specific classes of games, and generalizing them to wide classes (e.g., the first-cycle games of Aminof and Rubin are inspired by the paper of Ehrenfeucht and Mycielski on mean-payoff games [EM79]).

Complex specifications. Over the last decade, the increasing need to model complex specifications has shifted research toward games where multiple (quantitative and qualitative) objectives — often beyond ω-regular ones — interact, requiring the analysis of trade-offs between several objectives. In order to improve the understanding of these trade-offs, a lot of effort is put in studying games where objectives are rich Boolean combinations of objectives. We mention for example [CHP07] for combinations of parity, [CD12; CRR14; JLS15] for combinations of energy and parity, [VCD+15] for combinations of mean-payoff, [CDRR15] for combinations of total-payoff, or [BMR+18; BHM+17] for combinations of energy and average-energy objectives.

When considering such rich objectives, memoryless strategies usually do not suffice, and one has to use an amount of memory that can hinder the implementation (e.g., exponential memory) or that can prevent it (infinite memory). Establishing precise memory bounds for such general combinations of objectives is tricky and sometimes counterintuitive. For example, while mean-payoff games and energy games are memoryless-determined and inter-reducible in the single-objective setting, exponential-memory strategies are both sufficient and necessary for conjunctions of energy objectives [CRR14; JLS15] while *infinite-memory* strategies are required for conjunctions of mean-payoff ones [VCD+15].

[Sha53]: Shapley (1953), *Stochastic Games*

[EM79]: Ehrenfeucht et al. (1979), *Positional Strategies for Mean Payoff Games*

[CdHS03]: Chakrabarti et al. (2003), *Resource Interfaces*

[GZ04]: Gimbert et al. (2004), *When Can You Play Positionally?*

[GZ05]: Gimbert et al. (2005), *Games Where You Can Play Optimally Without Any Memory*

[CN06]: Colcombet et al. (2006), *On the positional determinacy of edge-labeled games*

[AR17]: Aminof et al. (2017), *First-cycle games*

[Kop06]: Kopczyński (2006), *Half-Positional Determinacy of Infinite Games*

[BFMM11]: Bianco et al. (2011), *Exploring the boundary of half-positionality*

[Ohl23]: Ohlmann (2023), *Characterizing Positionality in Games of Infinite Duration over Infinite Graphs*

[CHP07]: Chatterjee et al. (2007), *Generalized Parity Games*

[CD12]: Chatterjee et al. (2012), *Energy parity games*
[CRR14]: Chatterjee et al. (2014), *Strategy synthesis for multi-dimensional quantitative objectives*
[JLS15]: Jurdziński et al. (2015), *Fixed-Dimensional Energy Games are in Pseudo-Polynomial Time*

[VCD+15]: Velner et al. (2015), *The complexity of multi-mean-payoff and multi-energy games*

[CDRR15]: Chatterjee et al. (2015), *Looking at mean-payoff and total-payoff through windows*

[BMR+18]: Bouyer et al. (2018), *Average-energy games*
[BHM+17]: Bouyer et al. (2017), *Bounding Average-Energy Games*

1.4 Contributions

Our work starts from the observation that while there is a good understanding of memoryless determinacy in the literature (thanks to multiple characterizations and general sufficient conditions), this is not so much the case for finite-memory determinacy. Studying strategy complexity is a common staple in most works about zero-sum games on graphs, but there are few general results for strategies beyond memoryless ones. We give a high-level overview of our contributions, which are aimed at remedying this observation, and which we divide into two parts. More precise statements and comparisons with the literature will be given in introductory chapters to each part, as well as in the introduction of each contribution chapter.

1.4.1 Characterizing finite-memory determinacy

In the first part of the thesis, titled *Characterizing finite memory requirements*, our goal is to provide practical characterizations of finite-memory determinacy of game objectives. We strive for generality: we want our results to encompass as many objectives as possible. We will see that there are multiple kinds of *finite-memory determinacy*. In order to be more specific, we define what we mean by a *finite-memory strategy*: a finite-memory strategy consists of

▶ a *memory structure*: roughly, a deterministic automaton keeping track of some information based on the alphabet symbols already seen (the information it provides is given by its current state), and

▶ a *next-action function*, prescribing the actions to take depending on the graph vertex and the memory structure state.

In its more general meaning, finite-memory determinacy of an objective requires that in all graphs, both players can implement winning strategies for this objective with finite-memory strategies.

Arena-independent finite-memory determinacy. We identified a valuable strengthening of finite-memory determinacy which we called *arena-independent* finite-memory determinacy. This property of an objective requires that players can implement winning strategies with finite-memory strategies whose memory structure *can be taken as the same in all game graphs using this objective*. Such a memory structure can then depend on parameters of the objective, but not parameters of the game graph (called the *arena*) in which the game is played. The information that the memory structure can use to update its state can then not be elements observed in arenas (such as vertices and edges), but letters of the alphabet on which the objective is defined (this alphabet is called the set of *colors*). This leads to the convenient notion of *chromatic memory structures*, which can be instantiated without considering specific arenas. The idea of *chromatic memory*

structures was already developed by Kopczyński in his PhD thesis [Kop08]. Naturally, the winning strategies built on top of these chromatic memory structures will then depend on the actual game graphs for their next-action functions. Compared to the more general finite-memory determinacy, arena-independent finite-memory determinacy inverts the order of the quantifiers: there must exist a "uniform" memory structure such that in all graphs, there exist winning strategies using the memory structure.

Memoryless determinacy is a special case of arena-independent finite-memory determinacy: the memory structure used by the memoryless strategies is always the trivial structure with a single state. Arena-independent finite-memory determinacy, despite being less general than finite-memory determinacy, applies to many objectives, including all ω-regular ones. One of the main interests of this kind of determinacy is that we were able to extend (with some work) techniques usually applicable for the more precise *memoryless determinacy*. In particular, we generalized results about memoryless determinacy of games played on finite graphs by Gimbert and Zielonka [GZ05] and of games played on infinite graphs by Colcombet and Niwiński [CN06].

Finite game graphs, one-to-two-player lifts. For games on finite graphs, one of our main contributions is a practical result: to show arena-independent finite-memory determinacy of an objective, it suffices to show the property in the *one-player games* of both players. A one-player game is simply a game in which the same player controls all vertices — it is an edge-labeled graph. Finding a winning strategy in a one-player game (when it exists) simply consists of finding a winning path in the graph, with no need to take into account the opponent's behavior. This reasoning is *graph*-theoretic, which is simpler than corresponding *game*-theoretic reasonings for the same objective. We call such a result a *one-to-two-player lift*, as to understand (a kind of) strategy complexity of two-player zero-sum games, it suffices to understand the strategy complexity of one-player games, which is conceptually easier.

The first such result was formulated for memoryless determinacy of games played on finite graphs in [GZ05] (i.e., to show memoryless determinacy of an objective over finite graphs, it suffices to show the property in the *one-player games* on finite graphs of both players). Our result is an extension of it to the broader arena-independent finite-memory determinacy, with a relatively small increase in memory when going from one-player to two-player games: the *product* of the memory structures sufficient in the one-player games of each player suffices for both players in two-player games. Unfortunately, we also show that such a one-to-two-player lift for the more general finite-memory determinacy does not hold: there exists a game objective for which both players have finite-memory strategies (whose memory structure depends on the graph) in their one-player games, but infinite memory is required for a player in some two-player

[Kop08]: Kopczyński (2008), *Half-positional Determinacy of Infinite Games*

[GZ05]: Gimbert et al. (2005), *Games Where You Can Play Optimally Without Any Memory*

[CN06]: Colcombet et al. (2006), *On the positional determinacy of edge-labeled games*

game. Considering *arena-independent* finite-memory determinacy is therefore a natural restriction to preserve this powerful result. The lift is obtained through a characterization of arena-independent finite-memory determinacy using two language-theoretic properties.

Results similar to this *one-to-two-player lift* will be a common thread in our work, with alternate versions of it holding in other contexts, and with strengthenings available for more specific classes of objectives. We further the idea that strategy complexity of two-player zero-sum games often reduces to the simpler strategy complexity of one-player games, also developed by others [GZ05; Kop08; GZ09; Koz22b].

Infinite game graphs. A known fact about strategy complexity is that increasing the size of the game graphs may require using more complex strategies. In particular, some objectives are memoryless-determined when considered over finite graphs, but suddenly require infinite memory to implement winning strategy in some infinite game graph: this is the case for the *mean-payoff* objective [Put94].

There are therefore fewer objectives that admit "simple" winning strategies when taken over arbitrary graphs. On the other hand, the ever-present parity condition is an objective whose memoryless determinacy is preserved in infinite graphs (of any cardinality) [EJ91; Mos91; Zie98]. This is remarkable in its own way, but a converse of this result also holds, cementing the place of the parity condition as a special and unique objective. This converse, by Colcombet and Niwiński [CN06], states that any objective that is both memoryless-determined (over infinite graphs) and *prefix-independent* is a parity condition. *Prefix-independence* is a common technical assumption on objectives requiring that whether an infinite word is in the objective is not influenced by its finite prefixes; parity conditions and Muller conditions are prefix-independent.

Given our motivation to push results about strategy complexity from memoryless to finite-memory determinacy, a natural question becomes: *what objectives remain finite-memory determined over infinite game graphs?* We again focus on arena-independent finite-memory determinacy, both for technical convenience and as we do not know of an example distinguishing it from the more general finite-memory determinacy in games played on infinite graphs. We show that the objectives that are arena-independent-finite-memory-determined over infinite graphs are *exactly* the ω-regular objectives. One implication was known since [BL69; GH82] and was discussed above; the other implication is our main contribution in this characterization. It generalizes the results of Colcombet and Niwiński [CN06] in two ways: by generalizing the class of strategies and getting rid of the prefix-independence hypothesis. For the latter, we revisit the algebraic notion of *right congruence* at the core of the Myhill-Nerode theorem for languages of finite words [Ner58]. We obtain a precise link between the representation of an ω-regular objective as a deterministic

An overview of all the versions of this *one-to-two-player lift* is provided in Subsection 9.1.2 of the conclusion.

[GZ05]: Gimbert et al. (2005), *Games Where You Can Play Optimally Without Any Memory*
[Kop08]: Kopczyński (2008), *Half-positional Determinacy of Infinite Games*
[GZ09]: Gimbert et al. (2009), *Pure and Stationary Optimal Strategies in Perfect-Information Stochastic Games with Global Preferences*
[Koz22b]: Kozachinskiy (2022), *One-To-Two-Player Lifting for Mildly Growing Memory*

[Put94]: Puterman (1994), *Markov Decision Processes: Discrete Stochastic Dynamic Programming*

[EJ91]: Emerson et al. (1991), *Tree Automata, Mu-Calculus and Determinacy (Extended Abstract)*
[Mos91]: Mostowski (1991), *Games with Forbidden Positions*
[Zie98]: Zielonka (1998), *Infinite Games on Finitely Coloured Graphs with Applications to Automata on Infinite Trees*
[CN06]: Colcombet et al. (2006), *On the positional determinacy of edge-labeled games*

[BL69]: Büchi et al. (1969), *Definability in the Monadic Second-Order Theory of Successor*
[GH82]: Gurevich et al. (1982), *Trees, Automata, and Games*

[Ner58]: Nerode (1958), *Linear Automaton Transformations*

parity automaton based on its memory requirements and on its right congruence.

As a by-product, we once again obtain a one-to-two-player lift: it suffices to check arena-independent finite-memory determinacy of an objective in infinite one-player game graphs (of both players) to guarantee it in infinite two-player game graphs. However, compared to the previous lift for games played on finite graphs, there is a slightly greater memory increase when going from one-player to two-player games.

1.4.2 Precise memory requirements

We have up to now discussed characterizations of arena-independent finite-memory determinacy in various contexts. Despite being of wide applicability with respect to objectives, these results do not always provide *tight* memory requirements for each player; they usually only provide upper bounds on the size of the memory structures needed to implement optimal strategies. Moreover, these upper bounds do not distinguish the two players, even though their memory requirements can vary wildly. This is the case for the famous *Rabin conditions* [KK91], for which the first player does not need memory to implement winning strategies, but for which the second player, playing for a *Streett condition*, requires exponential memory [DJW97]. In the second part of the thesis, titled *Obtaining precise memory requirements*, we consider more precise classes of objectives in order to obtain more precise memory requirements.

[KK91]: Klarlund et al. (1991), *Rabin Measures and Their Applications to Fairness and Automata Theory*

[DJW97]: Dziembowski et al. (1997), *How Much Memory is Needed to Win Infinite Games?*

Missing pieces for ω-regular objectives.   Our choice for these classes stems from the following observation: *the memory requirements of ω-regular objectives are still not completely settled*. They are completely settled for the well-known class of Muller conditions through various characterizations [DJW97; Cas22; CCL22]; yet, this only provides an *upper bound* on the memory requirements of other ω-regular objectives. Indeed, an arbitrary ω-regular objective can be represented as a Muller condition only by adjoining it to an automatic structure (yielding a Muller automaton). The minimal memory structures that also take into account the automatic structures are not well-understood.

[Cas22]: Casares (2022), *On the Minimisation of Transition-Based Rabin Automata and the Chromatic Memory Requirements of Muller Conditions*
[CCL22]: Casares et al. (2022), *On the Size of Good-For-Games Rabin Automata and Its Link with the Memory in Muller Games*

Regular objectives.   To alleviate this, we first focus on what we called *regular reachability objectives*, which are the sets of infinite words having a finite prefix in a given regular language. In such games, the antagonistic opponent wants to prevent any finite prefix from being in the regular language for an infinite duration, which we call a *regular safety objective*. These objectives are very simple and can be represented by classical deterministic finite automata. Perhaps surprisingly, the minimal memory structures for these objectives were not yet understood, despite the simplicity of the problem (which can be rephrased as "*what is the minimal amount of*

information that must be remembered to realize a word from a regular language?").
Only the strategy complexity of regular safety objectives (for an alternate
memory model discussed in Chapter 2) was studied in [CFH14; CFH22].

We characterize the memory requirements of these objectives through
decidable language-theoretic properties (both for the player trying to
achieve the regular reachability objective and its opponent). We also prove
that, given a deterministic finite automaton as an input, it is NP-complete
to decide the existence of a sufficient memory structure with a given
number of states. In other words, it is not possible to find a smallest
memory structure in polynomial time unless P = NP. We implemented
the search for a smallest memory structure using a SAT solver.

In addition to these regular objectives, we also discuss the extension
of our results to *topologically open and closed objectives*, which are natural
generalizations of regular objectives through a topological point of view.

Deterministic Büchi automata. Despite being a novelty with regard to
memory requirements, regular objectives are still far from covering all
ω-regular objectives. To get closer to general ω-regular objectives, we con-
sidered automata giving rise to more complex objectives: the *deterministic
Büchi automata*. We do not show precise memory requirements, but we
characterize those recognizing objectives for which $\mathcal{P}_1$ has *memoryless*
winning strategies. This property is weaker than memoryless determinacy
as it is not required that $\mathcal{P}_2$ also has memoryless winning strategies, and
is often called *half-positionality* [Kop06; BFMM11]. Our characterization
consists of three conditions; two of them are inherited from the study
of regular objectives, and are still necessary conditions. However, their
conjunction is not sufficient anymore in this larger class of objectives. We
add a third condition (related to the right congruence and Myhill-Nerode
theorem) that was previously useful for the learning of languages [AFS20;
AF21], but that was not shown to be linked to strategy complexity. Together,
the three conditions are equivalent to half-positionality of objectives recog-
nizable by deterministic Büchi automata. This characterization is shown to
be *effective* and provides a way to decide the half-positionality of objectives
recognizable by deterministic Büchi automata in polynomial time.

Unlike memoryless determinacy, results about half-positionality were
mostly sufficient conditions and not characterizations [Kop06; BFMM11].
In particular, they do not encompass all half-positional objectives recog-
nized by deterministic Büchi automata. This situation changed recently
with a characterization of half-positionality over infinite game graphs
by Ohlmann [Ohl23], but this characterization is still difficult to apply
systematically to all ω-regular objectives. We use this new characterization
for one implication of our equivalence.

Precise lifts. As part of our characterizations for regular objectives and
deterministic Büchi automata, we obtain (for these specific classes of

[CFH14]: Colcombet et al. (2014), *Playing Safe*

[CFH22]: Colcombet et al. (2022), *Playing Safe, Ten Years Later*

[Kop06]: Kopczyński (2006), *Half-Positional Determinacy of Infinite Games*

[BFMM11]: Bianco et al. (2011), *Exploring the boundary of half-positionality*

[AFS20]: Angluin et al. (2020), *Polynomial Identification of ω-Automata*

[AF21]: Angluin et al. (2021), *Regular ω-languages with an informative right congruence*

[Ohl23]: Ohlmann (2023), *Characterizing Positionality in Games of Infinite Duration over Infinite Graphs*

objectives) one-to-two-player lifts stronger than the general ones from the first part. First, we show that the memory requirements of a single player in its own one-player games are exactly the same as in its two-player games (with no need to intertwine it with the memory requirements of the opponent). Secondly, we show that there is no difference between the memory requirements in finite and infinite game graphs. Combined, these results tell us that for these subclasses of ω-regular objectives, the memory requirements in one-player games on finite graphs are the same as in two-player games on graphs of arbitrary cardinality.

1.5 Outline

High-level structure. Each of the two parts of this thesis includes three chapters (respectively Chapters 3, 4, and 5 and Chapters 6, 7, and 8). The first chapter of each part consists in a motivation of the problems that the part addresses and the answers that it provides. It also contains a more precise discussion of the related literature focused on the problems at hand in addition to the references discussed above. The other two chapters in each part contain our main contributions. The chapters containing our contributions are all structured in a similar fashion, containing at least one section for each of the following items:

▸ notations and preliminaries about notions mainly used in the current chapter, complementing global preliminaries from Chapter 2;
▸ an overview of our results and contributions;
▸ detailed proofs of the results.

Chapter content. Chapter 2 contains the main preliminaries for the whole thesis: we formally introduce the model of games on graphs, some classical objectives, definitions of classes of strategies, and definitions of kinds of determinacy.

The next three chapters constitute the part *Characterizing finite memory requirements*.

▸ In Chapter 3, we introduce our contributions about characterizing objectives with finite memory requirements. We present existing works about memoryless determinacy, justify the relevance of *arena-independent finite-memory determinacy*, and sketch our results from the next two chapters.
▸ In Chapter 4, we characterize arena-independent finite-memory determinacy of games played on finite graphs. We provide both a language-theoretic characterization and a practical *one-to-two-player lift*, generalizing results from [GZ05]. Results from this chapter originate from a collaboration with Patricia Bouyer, Stéphane Le Roux, Youssouf Oualhadj, and Mickael Randour [BLO+20; BLO+22].

[GZ05]: Gimbert et al. (2005), *Games Where You Can Play Optimally Without Any Memory*
[BLO+20]: Bouyer et al. (2020), *Games Where You Can Play Optimally with Arena-Independent Finite Memory*
[BLO+22]: Bouyer et al. (2022), *Games Where You Can Play Optimally with Arena-Independent Finite Memory*

► In Chapter 5, we characterize arena-independent finite-memory determinacy of games played on arbitrary graphs. We find that objectives with this property are exactly the ω-regular ones, which provides a converse to the finite-memory determinacy of ω-regular objectives, and generalizes results from [CN06] about a characterization of memoryless determinacy. This allows relating the memory requirements of an objective with its representation as a deterministic parity automaton. Results from this chapter originate from a collaboration with Patricia Bouyer and Mickael Randour [BRV22a; BRV23].

The part *Obtaining precise memory requirements* is then contained in the next three chapters.

► In Chapter 6, we describe what is still unknown about the memory requirements of ω-regular objectives, despite the extensive literature on the topic. We argue that to go further, we need to understand memory requirements of the ω-regular objectives requiring representations with a non-trivial automaton structure.

► In Chapter 7, we introduce *regular objectives* as the subclass of the ω-regular ones definable with a simple deterministic finite automaton. We characterize precisely the memory requirements of regular objectives (for both players) and show that deciding the existence of small memory structures given an input automaton is NP-complete. Results from this chapter originate from a collaboration with Patricia Bouyer, Nathanaël Fijalkow, and Mickael Randour [BFRV22].

► In Chapter 8, we consider the broader class of ω-regular objectives recognizable by *deterministic Büchi automata*. We characterize exactly those for which $\mathcal{P}_1$ needs no memory to play optimally, and show that this property is decidable in polynomial time. Results from this chapter originate from a collaboration with Patricia Bouyer, Antonio Casares, and Mickael Randour [BCRV22].

In Chapter 9, we naturally end the thesis with a concluding part, summarizing our contributions and emphasizing some prospects and leads for the future.

Publication history.   Contributions in this thesis originate from the five published articles [BLO+20; BLO+22; BRV22a; BRV23; BCRV22] (journal articles [BLO+22; BRV23] respectively subsume the conference versions [BLO+20; BRV22a]), and the technical report [BFRV22] available online. An overview including five of the six articles [BLO+20; BLO+22; BRV22a; BRV23; BCRV22] detailed in this thesis was published as an 18-page invited paper in the proceedings of *FSTTCS'22* [BRV22b]. It can be seen as a short survey of the results of this thesis, excluding the results from [BFRV22] (Chapter 7) which are posterior to it. An additional work about stochastic games [BORV21a] is briefly discussed in Subsection 4.8.1 but not fully detailed.

[CN06]: Colcombet et al. (2006), *On the positional determinacy of edge-labeled games*

[BRV22a]: Bouyer et al. (2022), *Characterizing Omega-Regularity Through Finite-Memory Determinacy of Games on Infinite Graphs*
[BRV23]: Bouyer et al. (2023), *Characterizing Omega-Regularity Through Finite-Memory Determinacy of Games on Infinite Graphs*

[BFRV22]: Bouyer et al. (2022), *How to Play Optimally for Regular Objectives?*

[BCRV22]: Bouyer et al. (2022), *Half-Positional Objectives Recognized by Deterministic Büchi Automata*

[BLO+20]: Bouyer et al. (2020), *Games Where You Can Play Optimally with Arena-Independent Finite Memory*
[BLO+22]: Bouyer et al. (2022), *Games Where You Can Play Optimally with Arena-Independent Finite Memory*
[BRV22b]: Bouyer et al. (2022), *The True Colors of Memory: A Tour of Chromatic-Memory Strategies in Zero-Sum Games on Graphs (Invited Talk)*
[BORV21a]: Bouyer et al. (2021), *Arena-Independent Finite-Memory Determinacy in Stochastic Games*
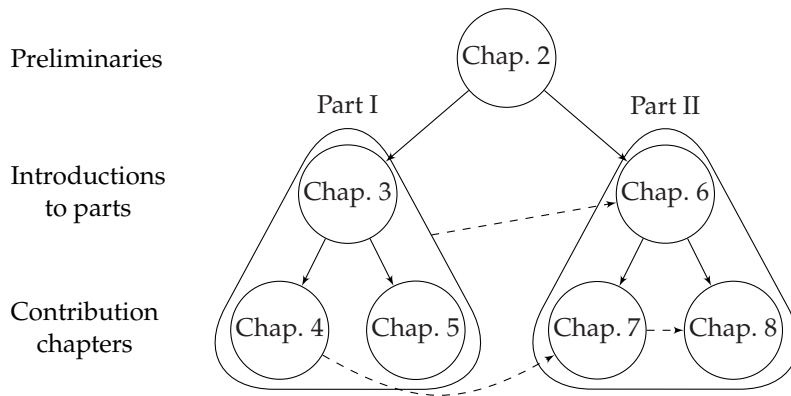
Figure 1.3: Structure and main dependencies between chapters of this thesis. Dashed dependencies are recommended to fully appreciate the technical arguments, but not necessary to grasp the results.

Reading tips. The contributions chapters are written to be readable as independently as possible; their results can be understood simply by reading the general preliminaries (Chapter 2) and the introductory chapter from their part. Naturally, links between chapters are still frequently established to make this document a consistent whole, but they should not prevent the reader from grasping the main contributions of each part. Notably, a proof in Chapter 7 uses a general theorem of Chapter 4, some motivations in Chapter 6 stem from the limits of the general results of the first part, and some technical arguments necessary for Chapter 8 already appear in Chapter 7. The dependencies between chapters are depicted in Figure 1.3.

If you ever feel lost, two features of this thesis may help you. First, Subsection 9.1.1 summarizes most properties defined in this thesis and their implications related to strategy complexity. Secondly, a table of notations encompassing all frequently used notations is available on page 257.

This thesis was composed using the LaTeX class kaobook. One of its main features is the ability to add comments in the margin, replacing footnotes. Comments are aligned horizontally with the content they refer to as much as possible. Short versions of the bibliographic references are also printed in the margin; a complete bibliography is of course available at the end of the document.

This class can be downloaded at https://github.com/fmarotta/kaobook.

Two-player turn-based games on graphs 2

This chapter introduces *turn-based games on graphs*, which constitute the main mathematical framework underlying this thesis. We focus on *two-player zero-sum games of infinite duration*, a prevalent model to describe the possibly never-ending interaction between a player seeking to achieve an objective and its antagonistic opponent. Notions and notations defined here are used throughout multiple chapters of this thesis. We start with some classical mathematical concepts and then move on to our models of games, objectives, strategies, and determinacy.

2.1 Mathematical notations

We respectively denote by $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$, and $\mathbb{R}$ the sets of natural numbers (starting at 0), integers, rational numbers, and real numbers.

Given a set A, we define A^* as the set of finite sequences of elements of A, A^+ as the set of non-empty finite sequences of elements of A, and A^ω as the set of infinite sequences of elements of A. The cardinality of set A is denoted by $|A|$. We denote the empty word by ε.

Let A be a set. A *preorder on A* (also called a *quasiorder*) is a binary relation $\leq \subseteq A \times A$ that is *reflexive* ($a \leq a$ for all $a \in A$) and *transitive* ($a_1 \leq a_2$ and $a_2 \leq a_3$ implies $a_1 \leq a_3$ for all $a_1, a_2, a_3 \in A$). Let $\leq$ be a preorder on A. We say that two elements $a_1, a_2 \in A$ are *comparable for $\leq$* if $a_1 \leq a_2$ or $a_2 \leq a_1$. Preorder $\leq$ is *total* if for all $a_1, a_2 \in A$, a_1 and a_2 are comparable. A set $\Gamma \subseteq A$ is a *chain for $\leq$* (resp. *antichain for $\leq$*) if for all $a_1, a_2 \in \Gamma$, a_1

and a_2 are comparable (resp. are not comparable) for $\leq$. A preorder $\leq$ is *well-founded* if every chain for $\leq$ contains a minimal element for $\leq$.

An *equivalence relation on A* is a preorder $\sim \subseteq A \times A$ that is additionally *symmetric* ($a_1 \sim a_2$ implies $a_2 \sim a_1$ for all $a_1, a_2 \in A$). For $\sim$ an equivalence relation and $a \in A$, we write $[a]_\sim = \{a' \in A \mid a \sim a'\}$ for the *equivalence class of a for* $\sim$. We write $A/_\sim$ for the *quotient of A by* $\sim$, i.e., the set of equivalence classes for $\sim$. The *index* of an equivalence relation is the cardinality of its set of equivalence classes.

2.2 Game arenas

Let C be a non-empty alphabet of *colors*. We assume throughout this document that there is always a quantified set C of colors, which is sometimes instantiated in examples.

We study zero-sum turn-based games on graphs involving two players interacting for an infinite duration. The two players are called $\mathcal{P}_1$ and $\mathcal{P}_2$. Players play on *arenas*, which are graphs such that each vertex is controlled by either $\mathcal{P}_1$ or $\mathcal{P}_2$. Intuitively, a configuration of the game is specified by some current vertex of the arena, and $\mathcal{P}_1$ (resp. $\mathcal{P}_2$) decides on the next vertex following an edge of the arena when the current vertex is controlled by $\mathcal{P}_1$ (resp. $\mathcal{P}_2$). We also label the edges of arenas with colors from C, which will be used later to specify game objectives. Observe that we take the convention that *edges* are colored, and not vertices (both formalisms are common in the literature).

> **Definition 2.2.1** *A (C-colored two-player) arena is a tuple $\mathcal{A} = (V, V_1, V_2, E)$ where V is a non-empty set of* vertices *such that $V = V_1 \uplus V_2$ (symbol $\uplus$ denotes a disjoint union) and $E \subseteq V \times C \times V$ is a set of (C-colored)* edges.

Let $\mathcal{A} = (V, V_1, V_2, E)$ be an arena. If $e = (v, c, v') \in E$, we denote v by $\mathsf{in}(e)$, c by $\mathsf{col}(e)$, and v' by $\mathsf{out}(e)$; i.e., for all edges $e \in E$, we have $e = (\mathsf{in}(e), \mathsf{col}(e), \mathsf{out}(e))$. Vertices in V_1 are controlled by $\mathcal{P}_1$ and vertices in V_2 are controlled by $\mathcal{P}_2$. We assume that arenas are *non-blocking* ("deadlock-free"), i.e., that for all vertices $v \in V$, there is at least one edge $e \in E$ such that $\mathsf{in}(e) = v$. In particular, $|E| \geq |V|$. This guarantees that the interaction between players can always be prolonged, as an edge will always be available in every vertex. This assumption is a technical convenience to avoid having to consider both finite and infinite interactions; in practice, a blocking vertex can be made non-blocking by adding a self-loop to it (this change may require tweaking the set of colors or the objective to preserve the expected winner in every situation).

An arena is *finite* if it has finitely many vertices and edges, is *countable* if it has countably many vertices and edges, and is *finitely branching* if for

Qualifiers between parentheses in definitions will usually not be recalled but always apply to the object being defined, unless otherwise specified. In this case, all arenas in this thesis are *C*-colored and two-player, and all edges are *C*-colored by default.
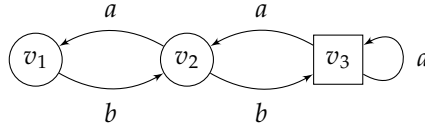
Figure 2.1: A finite arena $\mathcal{A}$. Circles are always used to depict vertices controlled by $\mathcal{P}_1$ (i.e., vertices in V_1) and squares for vertices controlled by $\mathcal{P}_2$ (i.e., vertices in V_2) (all graphical conventions are summed up on page 257).

all $v \in V$, there are finitely many edges $e \in E$ such that $\text{in}(e) = v$. Unless otherwise specified, arenas are allowed to be of any cardinality with no restriction on branching. Some results will only apply to restricted classes of arenas, but this will then be specified explicitly.

We give an example of an arena, which we also use to explain the graphical conventions to represent arenas.

Example 2.2.2 We consider an arena $\mathcal{A} = (V, V_1, V_2, E)$ depicted in Figure 2.1. This arena has three vertices ($V = \{v_1, v_2, v_3\}$), two of which are controlled by $\mathcal{P}_1$ ($V_1 = \{v_1, v_2\}$, depicted by circles) and one by $\mathcal{P}_2$ ($V_2 = \{v_3\}$, depicted by a square). It has five edges ($E = \{(v_1, b, v_2), (v_2, b, v_3), (v_3, a, v_3), (v_3, a, v_2), (v_2, a, v_1)\}$). Arena $\mathcal{A}$ is finite.

To describe finite and infinite interactions of $\mathcal{P}_1$ and $\mathcal{P}_2$ in an arena, we use respectively *histories* and *plays*.

Definition 2.2.3 *A* history *of arena $\mathcal{A} = (V, V_1, V_2, E)$ is a finite sequence $\gamma = e_1 \ldots e_n \in E^*$ such that for all i such that $1 \leq i \leq n - 1$, we have $\text{out}(e_i) = \text{in}(e_{i+1})$.*
History $\gamma = e_1 \ldots e_n \in E^*$ is from v *if $\text{in}(e_1) = v$.*

For convenience, we assume that for all vertices $v \in V$, there is a distinct *empty history* λ_v — this permits specifying an initial vertex even when no edge has been taken yet. The set of all histories of an arena $\mathcal{A}$ is denoted by $\text{Hists}(\mathcal{A})$. We extend functions in and out to histories in a natural way: if $\gamma = e_1 \ldots e_n$ is a history, we write $\text{in}(\gamma)$ for $\text{in}(e_1)$ and $\text{out}(\gamma)$ for $\text{out}(e_n)$. For an empty history λ_v, we assume that $\text{in}(\lambda_v) = \text{out}(\lambda_v) = v$.

For $V' \subseteq V$, we write $\text{Hists}(\mathcal{A}, V')$ for the set of histories γ of $\mathcal{A}$ such that $\text{in}(\gamma) \in V'$. For $\ell \in \{1, 2\}$ (indicating the index of a player), we write $\text{Hists}_\ell(\mathcal{A})$ for the set of histories γ on $\mathcal{A}$ such that $\text{out}(\gamma) \in V_\ell$: these are the histories that have reached a vertex of the arena controlled by $\mathcal{P}_\ell$.

Definition 2.2.4 *A* play *of arena $\mathcal{A} = (V, V_1, V_2, E)$ is an infinite sequence $\rho = e_1 e_2 \ldots \in E^\omega$ such that for all $i \geq 1$, $\text{out}(e_i) = \text{in}(e_{i+1})$.*
Play $\rho = e_1 e_2 \ldots \in E^\omega$ is from v *if $\text{in}(e_1) = v$.*

For a play $\rho = e_1 e_2 \ldots$, we once again define $\text{in}(\rho) = \text{in}(e_1)$. The set of all plays of arena $\mathcal{A}$ is denoted by $\text{Plays}(\mathcal{A})$. If $\gamma = e_1 \ldots e_n \in E^*$ is a history (resp. $\rho = e_1 e_2 \ldots \in E^\omega$ is a play), we write $\text{col}^*(\gamma)$ (resp. $\text{col}^\omega(\rho)$)

for the finite sequence $\mathsf{col}(e_1) \dots \mathsf{col}(e_n) \in C^*$ (resp. the infinite sequence $\mathsf{col}(e_1)\mathsf{col}(e_2) \dots \in C^{\omega}$).

Example 2.2.5 In the arena of Figure 2.1, the finite sequence $\gamma = (v_2, b, v_3)(v_3, a, v_3)(v_3, a, v_2)(v_2, a, v_1)$ is a history from v_2. Also, γ is in $\mathsf{Hists}_1(\mathcal{A})$ as it ends in v_1, which is a vertex of V_1, controlled by $\mathcal{P}_1$. We have $\mathsf{col}^*(\gamma) = baaa$.

The infinite sequence $\rho = (v_2, b, v_3)(v_3, a, v_3)^{\omega}$ (used as a shorthand for the infinite sequence $(v_2, b, v_3)(v_3, a, v_3)(v_3, a, v_3) \dots$ with (v_3, a, v_3) repeated ad infinitum) is a play of $\mathcal{A}$ from v_2 such that $\mathsf{col}^{\omega}(\rho) = ba^{\omega}$.

We will often study a special kind of arenas in which the same player controls all the vertices.

Definition 2.2.6 *An arena* $\mathcal{A} = (V, V_1, V_2, E)$ *is a* one-player arena of $\mathcal{P}_1$ *(resp. of* $\mathcal{P}_2$*) if* $V_2 = \emptyset$ *(resp.* $V_1 = \emptyset$*).*

Such arenas are often easier to study than "two-player" arenas, as there is no game-theoretic reasoning involved: in a one-player arena of $\mathcal{P}_{\ell}$, $\mathcal{P}_{\ell}$ can decide alone what play is produced with no influence from its opponent.

2.3 Strategies

We specify the behavior of each player using *strategies*. A strategy describes how a player behaves in any situation in which this player has to react.

Let $\mathcal{A} = (V, V_1, V_2, E)$ be an arena and $\ell \in \{1, 2\}$ be the index of a player. Player $\mathcal{P}_{\ell}$ has to play a move every time the current history reaches a vertex controlled by $\mathcal{P}_{\ell}$, i.e., when this history is in $\mathsf{Hists}_{\ell}(\mathcal{A})$. In such a situation, the only constraint for the move chosen by $\mathcal{P}_{\ell}$ (consisting in the next edge in E to follow) is that it starts from the current vertex. We formalize these observations in the following definition.

Definition 2.3.1 *A strategy of* $\mathcal{P}_{\ell}$ *on* $\mathcal{A}$ *is a function* $\sigma_{\ell} \colon \mathsf{Hists}_{\ell}(\mathcal{A}) \to E$ *such that for all* $\gamma \in \mathsf{Hists}_{\ell}(\mathcal{A})$*,* $\mathsf{out}(\gamma) = \mathsf{in}(\sigma_{\ell}(\gamma))$*.*

A strategy is usually denoted by symbol σ, and the occasional $\ell \in \{1, 2\}$ subscript is simply used to help the reader remember to which player this strategy belongs.

Once a strategy of $\mathcal{P}_{\ell}$ is fixed, there may yet be multiple possible plays, depending on the strategy used by the opponent of $\mathcal{P}_{\ell}$. These plays are the ones that are *consistent* with the strategy.

Definition 2.3.2 *Given a strategy* σ_{ℓ} *of* $\mathcal{P}_{\ell}$ *on* $\mathcal{A}$*, a play* $\rho = e_1 e_2 \dots \in \mathsf{Plays}(\mathcal{A})$ *is consistent with* σ_{ℓ} *if for all finite prefixes* $\gamma = e_1 \dots e_i$ *of* ρ *such that* $\mathsf{out}(\gamma) \in V_{\ell}$*,* $\sigma_{\ell}(\gamma) = e_{i+1}$*.*

For $v \in V$, we denote by $\mathsf{Plays}(\mathcal{A}, v, \sigma_{\ell})$ the set of plays on $\mathcal{A}$ from v that are consistent with σ_{ℓ}. Notice that when $\mathcal{A}$ is a one-player arena of $\mathcal{P}_{\ell}$, the

set $\mathsf{Plays}(\mathcal{A}, v, \sigma_\ell)$ is a singleton. Indeed, as the opponent of $\mathcal{P}_\ell$ has then no choice to make, a strategy of $\mathcal{P}_\ell$ always determines a single play from v in a deterministic manner. We write $\mathsf{Plays}(\mathcal{A}, v, \sigma_1, \sigma_2)$ for the singleton set containing the unique play consistent with a couple of strategies σ_1, σ_2 of the two players (always writing the strategy of $\mathcal{P}_1$ first).

The set of plays consistent with a strategy σ_ℓ coincides exactly with the set of the plays that the opponent can induce against this strategy.

Lemma 2.3.3 *Let $\mathcal{A} = (V, V_1, V_2, E)$ be an arena, $v \in V$, and σ_1 be a strategy of $\mathcal{P}_1$ on $\mathcal{A}$. We have*

$$\mathsf{Plays}(\mathcal{A}, v, \sigma_1) = \bigcup_{\text{strategy } \sigma_2 \text{ of } \mathcal{P}_2 \text{ on } \mathcal{A}} \mathsf{Plays}(\mathcal{A}, v, \sigma_1, \sigma_2).$$

Proof. Any play that is consistent with both σ_1 and σ_2 is in particular consistent with σ_1, which shows the right-to-left inclusion.

Reciprocally, let $\rho = e_1 e_2 \ldots \in \mathsf{Plays}(\mathcal{A}, v, \sigma_1)$. We define a strategy σ_2 of $\mathcal{P}_2$ on $\mathcal{A}$ such that, when $e_1 \ldots e_i \in \mathsf{Hists}_2(\mathcal{A})$, $\sigma_2(e_1 \ldots e_i) = e_{i+1}$. Strategy σ_2 is defined arbitrarily on the other histories. By construction, $\mathsf{Plays}(\mathcal{A}, v, \sigma_1, \sigma_2) = \{\rho\}$. $\square$

Remark 2.3.4 We consider here *pure* strategies, that is, strategies that do not resort to randomization. In some contexts, it makes sense to define strategies as functions $\mathsf{Hists}_\ell(\mathcal{A}) \to \mathsf{Dist}(E)$, where $\mathsf{Dist}(E)$ denotes the set of distributions over E. We can then wonder about the existence of almost surely winning rather than surely winning strategies. There are many interesting questions about this alternate model, but this is not the central focus of this thesis — we will briefly mention some of these questions in Section 4.8.

2.4 Objectives and games

In order to make games interesting, we need to give each player a purpose. We do so by defining *game objectives*.

An objective is simply a way to describe the outcomes or behaviors that are deemed favorable by $\mathcal{P}_1$: it is a set of infinite words on alphabet C, i.e., a subset of C^ω. For such an objective, $\mathcal{P}_1$ wins when the infinite word resulting from the infinite interaction with $\mathcal{P}_2$ is in this set. As games are zero-sum, the objective of $\mathcal{P}_2$ is then to obtain an infinite sequence of colors *not* in this set. A game endowed with an objective, after the two players have interacted for an infinite duration, is either won by $\mathcal{P}_1$ and lost by $\mathcal{P}_2$, or won by $\mathcal{P}_2$ and lost by $\mathcal{P}_1$. Such objectives are called *qualitative* (or sometimes *Boolean* or *win-lose*), as they are either won or lost by a player.

> **Definition 2.4.1** *A (qualitative) objective is a set $W \subseteq C^\omega$.*

A more general way to encode a purpose for the players is to define a preorder on C^ω, which specifies which infinite words are preferable for $\mathcal{P}_1$. This more general formalism will only be used in Chapter 4, so we defer its introduction to this moment.

Given an objective W, we write $\overline{W} = C^\omega \setminus W$ for its *complement*. When an objective W is clear in the context, a word $w \in W$ is called *winning* (for $\mathcal{P}_1$), while a word $w \in \overline{W}$ is called *losing* (for $\mathcal{P}_1$).

The reason to define objectives using colors and not directly using vertices or edges of arenas (as is also often done) is that it allows asking questions about objectives (e.g., what properties does it satisfy?) with no need to instantiate arenas.

A *game* is then simply the combination of an arena (describing the possible interactions between the players) and an objective (describing the goal of $\mathcal{P}_1$). The goal of the antagonistic opponent $\mathcal{P}_2$ is implicit and is the complement of this objective, as we consider zero-sum games.

> **Definition 2.4.2** *A (qualitative) game is a tuple $\mathcal{G} = (\mathcal{A}, W)$ where $\mathcal{A}$ is an arena and W is an objective.*

We refer to a game as *one-player* if it is played on a one-player arena.

2.4.1 Optimality and determinacy

Let $\mathcal{G} = (\mathcal{A} = (V, V_1, V_2, E), W)$ be a game, and $v \in V$ be a vertex of the underlying arena.

> **Definition 2.4.3** *A strategy σ_1 of $\mathcal{P}_1$ on $\mathcal{A}$ is* winning from v for $\mathcal{P}_1$ in $\mathcal{G}$ *if for all $\rho \in \mathsf{Plays}(\mathcal{A}, v, \sigma_1)$, $\mathsf{col}^\omega(\rho) \in W$.*

This definition of winning strategy simply means that no matter how $\mathcal{P}_2$ behaves, σ_1 guarantees the win of $\mathcal{P}_1$. When the objective W is clear from the context, we often say that a strategy is *winning from v for $\mathcal{P}_1$ in $\mathcal{A}$*.

We define a corresponding notion for the opponent $\mathcal{P}_2$: we say that a strategy σ_2 of $\mathcal{P}_2$ on $\mathcal{A}$ is *winning from v for $\mathcal{P}_2$ in $\mathcal{G}$* if for all $\rho \in \mathsf{Plays}(\mathcal{A}, v, \sigma_2)$, $\mathsf{col}^\omega(\rho) \in \overline{W}$. This is the same definition, but using the complement objective $\overline{W}$.

The *winning region of a player in $\mathcal{G}$* is the set of vertices of $\mathcal{A}$ from which this player has a winning strategy.

Given a game and an arena vertex, the players cannot both have a winning strategy from this vertex. Yet, a property that is not immediate is that at least one player has a winning strategy from the vertex; indeed, every

strategy of a player could be "countered" by a strategy of its opponent. When an objective admits a winning strategy for either of the players in all arenas from all vertices, we say that it is *determined*.

> **Definition 2.4.4** *An objective W is* determined *if for all arenas $\mathcal{A} = (V, V_1, V_2, E)$, for all vertices $v \in V$, either $\mathcal{P}_1$ or $\mathcal{P}_2$ has a winning strategy from v in game $\mathcal{G} = (\mathcal{A}, W)$.*

Martin [Mar75] showed the very general result that all *Borel objectives* are determined. We do not define the Borel hierarchy here. All the objectives considered throughout this thesis are Borel (and are usually quite low on the Borel hierarchy) and are therefore determined.

As a convention, we will not only be looking for strategies that win from a single vertex, but that win from every vertex of the winning region. We call such strategies *optimal*, which is sometimes called *uniformly winning* or *uniformly optimal*.

> **Definition 2.4.5** *For $\ell \in \{1, 2\}$, a strategy of $\mathcal{P}_\ell$ is* (uniformly) optimal *for $\mathcal{P}_\ell$ in $\mathcal{G} = (\mathcal{A}, W)$ if it is winning from all the vertices of $\mathcal{A}$ from which $\mathcal{P}_\ell$ has a winning strategy, i.e., from all the vertices of the winning region of $\mathcal{P}_\ell$.*

We often write *optimal for $\mathcal{P}_\ell$ in $\mathcal{A}$* if the objective is clear from the context.

> **Remark 2.4.6** We stress that this notion of optimality requires a *single* strategy to be winning from *all* the winning vertices (even when we do not specify "uniformly" in what follows). In general, as strategies may observe the initial vertex, a player can always build an optimal strategy by taking the "union" of the winning strategies for each vertex of its winning region.
>
> However, asking for uniformity may require strategies that are *more complex to implement* than if we just required winning strategies from individual vertices (classes of complexity of strategies will be defined in Section 2.5). Uniformity is a common requirement (see, e.g., [GZ05; KMS+20; Ohl23]), that comes at no extra cost in many well-studied situations [McN93; DJW97; CN06]. We will come back to this definition and discuss in more depth what is the (mostly small) difference of requiring uniformity in Section 4.7 of Chapter 4, when we are armed with more tools.

2.4.2 Common examples of objectives

We give some examples of simple objectives. We will often use these as building blocks to study more complex objectives.

We first define two objectives that simply consist of seeing a specified color *once* or *infinitely many times*.

[Mar75]: Martin (1975), *Borel determinacy*

Formally, Martin considers the slightly different formalism of *Gale-Stewart* games [GS53], which are specified by an objective but not played on an arena. Still, the additional constraints from an arena (even an infinite one) and an initial vertex can be encoded into the objective while preserving its Borel property.

[GS53]: Gale et al. (1953), *Infinite Games with Perfect Information*

[GZ05]: Gimbert et al. (2005), *Games Where You Can Play Optimally Without Any Memory*
[KMS+20]: Kiefer et al. (2020), *How to Play in Infinite MDPs (Invited Talk)*
[Ohl23]: Ohlmann (2023), *Characterizing Positionality in Games of Infinite Duration over Infinite Graphs*

[McN93]: McNaughton (1993), *Infinite Games Played on Finite Graphs*
[DJW97]: Dziembowski et al. (1997), *How Much Memory is Needed to Win Infinite Games?*
[CN06]: Colcombet et al. (2006), *On the positional determinacy of edge-labeled games*
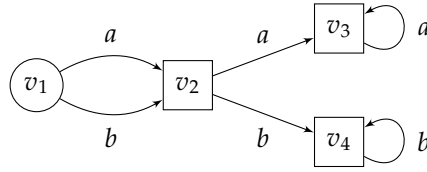
Definition 2.4.7 *If $a \in C$, the* reachability condition Reach(a) $\subseteq C^\omega$ *is the set of infinite words that see a at least once; i.e.,*

$$Reach(a) = \{c_1 c_2 \ldots \in C^\omega \mid \exists i \geq 1, c_i = a\}.$$

The complement of a reachability condition is called a safety condition. *The safety condition $\underline{Safe(a)}$ is the objective consisting of the infinite words that see no a, i.e., $\overline{Reach(a)}$.*

Definition 2.4.8 *If $a \in C$, the* Büchi condition Büchi(a) $\subseteq C^\omega$ *is the set of infinite words that see a infinitely often; i.e.,*

$$Büchi(a) = \{c_1 c_2 \ldots \in C^\omega \mid \forall j \geq 1, \exists i \geq j, c_i = a\}.$$

Example 2.4.9 Let $C = \{a, b\}$. In the arena of Figure 2.2, the winning region of $\mathcal{P}_1$ is $\{v_1, v_3\}$ for objective Reach(a), $\{v_4\}$ for objective Safe(a), and $\{v_3\}$ for objective Büchi(a). Interestingly, $\mathcal{P}_1$ cannot win for objective $W = $ Reach(a)∩Reach(b) (even from v_1), as $\mathcal{P}_2$ may observe the first edge chosen by $\mathcal{P}_1$ and take the edge with the same color in v_2, guaranteeing that $\mathcal{P}_1$ will not see both a and b.

A central objective for reactive synthesis (for instance, as any linear-time temporal logic specification [Pnu77] can be reduced to it with some work) is the *parity condition*.

Definition 2.4.10 *If $C = \{0, \ldots, n\}$ for some $n \in \mathbb{N}$, the* parity condition Parity(n) $\subseteq C^\omega$ *is the set of infinite words such that the maximal color they see infinitely often is even, i.e.,*

$$Parity(n) = \{c_1 c_2 \ldots \in C^\omega \mid \limsup_{i \to \infty} c_i \text{ is even}\}.$$

Some authors also use lim inf to define the parity condition, but we opt for the lim sup convention throughout the document.

Note that as C is finite for parity conditions, $\limsup_{i \to \infty} c_i$ is always a well-defined natural number between 0 and n.

The parity condition is concerned about the largest color seen infinitely often. It is a special case of a *Muller condition*, which is concerned about which colors are seen infinitely often.

Definition 2.4.11 *Let C be a finite set of colors, and $\mathcal{F} \subseteq 2^C$. The* Muller condition $\mathsf{Muller}(\mathcal{F}) \subseteq C^\omega$ *is the set of infinite words whose set of colors seen infinitely often is exactly an element of $\mathcal{F}$, i.e.,*

$$\mathsf{Muller}(\mathcal{F}) = \{w \in C^\omega \mid \{c \in C \mid w \in \mathsf{B\ddot{u}chi}(c)\} \in \mathcal{F}\}.$$

Notation 2^C refers to the power set of C.

We also define a *mean-payoff objective*. Taking colors as numbers, we look at the "average color seen" of an infinite word; formally, we consider the limit of the averages of the finite prefixes. We take the limit superior to avoid definability issues. We assume that $\mathcal{P}_1$ wants to keep the average above some threshold, which we take to be 0.

For objectives considering the value of some quantity, we take the convention that $\mathcal{P}_1$ prefers high values and $\mathcal{P}_2$ prefers low values.

Definition 2.4.12 *For $C \subseteq \mathbb{Q}$, the* mean-payoff function *is the function* $\mathsf{MP} \colon C^\omega \to \mathbb{R}$ *such that*

$$\mathsf{MP}(c_1 c_2 \ldots) = \limsup_{n \to \infty} \frac{1}{n} \sum_{i=1}^{n} c_i.$$

The mean-payoff objective (with threshold 0) *is then defined as*

$$\mathsf{MP}^{\geq 0} = \{w \in C^\omega \mid \mathsf{MP}(w) \geq 0\}.$$

We will see other ways to specify complex objectives using *automata* in Section 2.7.

2.5 Classes of simple strategies

A strategy of $\mathcal{P}_\ell$, which is a function $\sigma_\ell \colon \mathsf{Hists}_\ell(\mathcal{A}) \to E$, may be hard to describe: the set $\mathsf{Hists}_\ell(\mathcal{A})$ is usually infinite (even when $\mathcal{A}$ is finite), and there is no guarantee that there is a nice and finite description of σ_ℓ. Moreover, in practical applications (e.g., for synthesis), even in cases where a strategy can be described in a finite way, it may still be hard to implement.

Our goal here is to understand the contexts (which players, which objectives, which arenas?) for which *simple* strategies suffice to play optimally — in other words, for which there is no need to resort to complex, hard-to-implement strategies in order to win when winning is possible. We therefore need to define some notion of *complexity* of strategies. We use here a standard notion of complexity of strategies based on a simple automata-theoretic model. It is roughly similar to *Mealy machines* [Mea55], which are deterministic finite automata storing information, with additional outputs that can be used to describe the action taken by the players in the arenas. Syntactically, we split the Mealy machines into these two parts: the *memory structure* describes the deterministic finite automata, and the *next-action function* describes the action of the players. When a strategy can be implemented with a (finite) Mealy machine, we say that it

[Mea55]: Mealy (1955), *A method for synthesizing sequential circuits*

can be implemented with *finite memory*. We can then classify strategies in multiple classes that are more or less difficult to implement based on the answers to the following questions:

▶ is it possible to implement it with finite memory?
▶ if yes, how many states are needed to implement it as such?

2.5.1 Finite-memory strategies

We split the definition of a strategy with memory into two components: the underlying automatic structure on the alphabet of colors (called *memory structure*) and the output function describing actions in arenas (called *next-action function*).

> **Definition 2.5.1** *A* (chromatic) memory structure *is a tuple* $\mathcal{M} = (M, m_{\text{init}}, \alpha_{\text{upd}})$ *where M is a finite set of* states, $m_{\text{init}} \in M$ *is an* initial state, *and* $\alpha_{\text{upd}} \colon M \times C \to M$ *is a (deterministic, complete)* update function.

A chromatic memory structure is syntactically almost the same as a complete and deterministic finite automaton, except that we do not specify final states. The memory structure tracks what happens in games for an infinite duration, updating its state every time an edge is taken by observing the color of the edge. It is called *chromatic* as it observes the *colors* seen in the game. We extend notation α_{upd} to sequences of colors in C^* in a natural way: we define $\alpha_{\text{upd}}^* \colon M \times C^* \to M$ by induction on the length of finite words to be such that

We will discuss a relaxation of the *chromatic* requirement in Section 2.6, but this chromatic memory model is our main memory model.

▶ for all $m \in M$, $\alpha_{\text{upd}}^*(m, \varepsilon) = m$,
▶ inductively, for all $m \in M$ and $wc \in C^* \times C$, $\alpha_{\text{upd}}^*(m, wc) = \alpha_{\text{upd}}(\alpha_{\text{upd}}^*(m, w), c)$.

We recall that ε is the empty word.

A memory structure is used to store information: its state is updated depending on the colors that are seen while playing the game, and the current state is used to provide information in order to make decisions. In order to make a memory structure into a finite-memory strategy, we define a *next-action function* on top of it.

> **Definition 2.5.2** *For $\mathcal{M} = (M, m_{\text{init}}, \alpha_{\text{upd}})$ a memory structure, a strategy σ_ℓ of $\mathcal{P}_\ell$ on arena $\mathcal{A}$ is* based on (memory) $\mathcal{M}$ *if there exists a* next-action function
> $$\alpha_{\text{nxt}} \colon V_\ell \times M \to E$$
> *such that for all histories $\gamma \in \text{Hists}_\ell(\mathcal{A})$,*
> $$\sigma_\ell(\gamma) = \alpha_{\text{nxt}}(\text{out}(\gamma), \alpha_{\text{upd}}^*(m_{\text{init}}, \text{col}^*(\gamma))). \tag{2.1}$$

In other words, this definition requires that the decision made by strategy σ_ℓ always corresponds to the decision of α_{nxt} for the current arena vertex and the current memory state. For conciseness, when $\mathcal{M}$ is clear in the

Figure 2.3: Arena (left) and memory structure (right) used in Example 2.5.3. In figures, we always use diamonds (i.e., rhombuses) for memory states. The state with the incoming arrow denotes the initial state.

context, we sometimes abusively assume that a strategy of $\mathcal{P}_\ell$ based on $\mathcal{M}$ is this next-action function $\alpha_{\mathsf{nxt}}\colon V_\ell \times M \to E$.

> **Example 2.5.3** Let $C = \{a, b\}$. We consider objective $W = \mathsf{Büchi}(a) \cap \mathsf{Büchi}(b)$ consisting of the infinite words that see both a and b infinitely often (W is also equal to $\mathsf{Muller}(\{\{a, b\}\})$). In the arena $\mathcal{A} = (V = \{v\}, V_1 = \{v\}, V_2 = \emptyset, E = \{(v, a, v), (v, b, v)\})$ of Figure 2.3 (left), $\mathcal{P}_1$ has many winning strategies from v that may be complex to express (for instance, $\mathcal{P}_1$ wins by inducing word $abab^2ab^3\ldots$).
>
> Yet, $\mathcal{P}_1$ also has a winning strategy using the memory structure $\mathcal{M}$ with only two states depicted on the right of the figure. This memory structure remembers whether a or b was just seen throughout a game thanks to its two states: in m_1, either the game just started or a was just seen, and in m_2, b was just seen. Using next-action function $\alpha_{\mathsf{nxt}}\colon V_1 \times \{m_1, m_2\} \to E$ defined by $\alpha_{\mathsf{nxt}}(v, m_1) = (v, b, v)$ and $\alpha_{\mathsf{nxt}}(v, m_2) = (v, a, v)$, $\mathcal{P}_1$ induces the winning word $(ba)^\omega$.
>
> For this objective, upcoming results will imply that in *any* arena (not only in this simple one with a single vertex), this memory structure with two states suffices to implement optimal strategies for $\mathcal{P}_1$.

Strategies based on a memory structure respect the intuition that "adding information is never detrimental". One way to add information is to take the *(direct) product* of a memory structure with another one.

> **Definition 2.5.4** Let $\mathcal{M}_1 = (M_1, m^1_{\mathsf{init}}, \alpha^1_{\mathsf{upd}})$ and $\mathcal{M}_2 = (M^2, m^2_{\mathsf{init}}, \alpha^2_{\mathsf{upd}})$ be two memory structures. We define their product $\mathcal{M}_1 \otimes \mathcal{M}_2$ as the memory structure $\mathcal{M} = (M, m_{\mathsf{init}}, \alpha_{\mathsf{upd}})$ obtained as follows: $M = M_1 \times M_2$, $m_{\mathsf{init}} = (m^1_{\mathsf{init}}, m^2_{\mathsf{init}})$, and, for all $m_1 \in M_1$, $m_2 \in M_2$, $c \in C$, $\alpha_{\mathsf{upd}}((m_1, m_2), c) = (\alpha^1_{\mathsf{upd}}(m_1, c), \alpha^2_{\mathsf{upd}}(m_2, c))$.

That is, the memory structures are updated in parallel when a color is read. The same strategies can always be played with more information.

> **Lemma 2.5.5** Let σ_ℓ be a strategy of $\mathcal{P}_\ell$ on $\mathcal{A}$. If σ_ℓ is based on a memory structure $\mathcal{M}$, then for all memory structures $\mathcal{M}'$, σ_ℓ is also based on $\mathcal{M} \otimes \mathcal{M}'$.

Proof. We write $\mathcal{A} = (V, V_1, V_2, E)$. We assume that σ_ℓ is based on $\mathcal{M} = (M, m_{\mathsf{init}}, \alpha_{\mathsf{upd}})$, with next-action function $\alpha_{\mathsf{nxt}}\colon V_\ell \times M \to E$. Let $\mathcal{M}' = (M', m'_{\mathsf{init}}, \alpha'_{\mathsf{upd}})$. We simply define the next-action function $\alpha'_{\mathsf{nxt}}\colon V_\ell \times (M \times M') \to E$ that ignores the extra information given by $\mathcal{M}'$, i.e., such

that $\alpha'_{\mathsf{nxt}}(v, (m, m')) = \alpha_{\mathsf{nxt}}(v, m)$. This next-action function induces the same strategy as σ_ℓ and witnesses that σ_ℓ is also based on $\mathcal{M} \otimes \mathcal{M}'$. $\quad\square$

2.5.2 Memoryless strategies

We discuss an interesting subclass of finite-memory strategies that are arguably the simplest possible strategies. We let

$$\mathcal{M}_{\mathsf{triv}} = (\{m_{\mathsf{init}}\}, m_{\mathsf{init}}, (m_{\mathsf{init}}, c) \mapsto m_{\mathsf{init}})$$

denote the only *"trivial" memory structure* with a single state.

> **Definition 2.5.6** *A strategy is* memoryless *if it is based on the trivial memory structure* $\mathcal{M}_{\mathsf{triv}}$.

For conciseness, we sometimes abusively assume that a memoryless strategy of $\mathcal{P}_\ell$ is a function $V_\ell \to E$ (exactly as we did above for function α_{nxt}, except that we omit to specify set M since it is a singleton).

Memory structure $\mathcal{M}_{\mathsf{triv}}$ can be simply depicted as
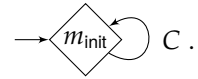
2.6 Flavors of finite-memory determinacy

We aim to understand the objectives for which finite-memory strategies suffice to play optimally and, when that is the case, how complex the strategies must be. In this section, we discuss precisely what it means.

2.6.1 Memoryless determinacy

We start with the simple memoryless strategies. In what follows, every variant of determinacy can be instantiated on multiple classes of arenas; common ones are the classes of finite arenas, countable arenas, finitely branching arenas, one-player arenas, or combinations of these qualifiers.

> **Definition 2.6.1** *An objective W is* memoryless-determined (over resp. finite, countable, finitely branching, one-player arenas) *if for all (resp. finite, countable, finitely branching, one-player) arenas $\mathcal{A}$, $\mathcal{P}_1$ and $\mathcal{P}_2$ have a memoryless optimal strategy in game $(\mathcal{A}, W)$.*

A common term in the literature to express that $\mathcal{P}_1$ (but not necessarily $\mathcal{P}_2$) has memoryless optimal strategies for an objective is *half-positionality* of the objective [Kop06] (*positional* is often used in the literature for *memoryless*). We will use this terminology in Chapter 8.

[Kop06]: Kopczyński (2006), *Half-Positional Determinacy of Infinite Games*

Remarkably, memoryless determinacy, despite being a very strong kind of determinacy, holds for many classical objectives.

> **Theorem 2.6.2** (Folklore) *Reachability, safety, and Büchi conditions are memoryless-determined.*

See [BCJ18] for proofs of memoryless determinacy for these objectives.

For synthesis, perhaps the most well-known and useful objective is the parity condition, which is partly due to its memoryless determinacy, first proved by Emerson, Jutla, and Mostowski [EJ91; Mos91].

> **Theorem 2.6.3** ([EJ91; Mos91]) *Parity conditions are memoryless-determined.*

Note that for the above examples, memoryless determinacy holds over arenas of arbitrary cardinality.

On the other hand, the mean-payoff objective $\mathsf{MP}^{\geq 0}$ is memoryless-determined over finite arenas [EM79], but not over arbitrary arenas [Put94]. We will see a proof of the memoryless determinacy of $\mathsf{MP}^{\geq 0}$, as well as an example of an infinite arena in which $\mathcal{P}_1$ can win but not with a memoryless strategy in Section 3.1.

> **Theorem 2.6.4** ([EM79]) *The mean-payoff objective is memoryless-determined over finite arenas.*

2.6.2 Chromatic versus chaotic memory structures

Our current *chromatic* memory model is actually slightly more restrictive than what is often used in the literature, because it can only observe *colors*. The following model, called *chaotic* [Kop08] (or sometimes *general* [Cas22]) can also observe the edges taken in the arenas.

> **Definition 2.6.5** *Let* $\mathcal{A} = (V, V_1, V_2, E)$ *be an arena. A* chaotic memory structure for $\mathcal{A}$ *is a tuple* $\mathfrak{M} = (M, m_{\mathsf{init}}, \alpha_{\mathsf{upd}})$ *where* M *is a finite set of states,* $m_{\mathsf{init}} \in M$ *is an* initial *state, and* $\alpha_{\mathsf{upd}} \colon M \times E \to M$ *is a (deterministic, complete) update function.*

The only difference with the chromatic memory model presented above is in the update function; $\alpha_{\mathsf{upd}} \colon M \times E \to M$ observes not just the colors, but also the full edges. As edges contain the color seen (they are elements in $V \times C \times V$), observing edges is syntactically more powerful. In practice, this means that during a game, the *chromatic* memory model only observes the sequence of *colors* seen, whereas the *chaotic* memory model observes the sequence of *edges* seen (i.e., the current memory state is determined by the word in C^* for the former and by the history in E^* for the latter). The definition of *strategy based on* $\mathcal{M}$ (Definition 2.5.2) also works with a chaotic memory structure $\mathfrak{M}$ for an arena by adjoining it to a next-action function.

[BCJ18]: Bloem et al. (2018), *Graph Games and Reactive Synthesis*

[EJ91]: Emerson et al. (1991), *Tree Automata, Mu-Calculus and Determinacy (Extended Abstract)*
[Mos91]: Mostowski (1991), *Games with Forbidden Positions*

[EM79]: Ehrenfeucht et al. (1979), *Positional Strategies for Mean Payoff Games*
[Put94]: Puterman (1994), *Markov Decision Processes: Discrete Stochastic Dynamic Programming*

[Kop08]: Kopczyński (2008), *Half-positional Determinacy of Infinite Games*

[Cas22]: Casares (2022), *On the Minimisation of Transition-Based Rabin Automata and the Chromatic Memory Requirements of Muller Conditions*

The only difference is in Equation (2.1), where $\mathsf{col}^*(\gamma)$ should be replaced by γ.

These two different memory models give rise to two distinct notions of *finite-memory determinacy*. The most general one (which we simply call *finite-memory determinacy*) uses the more powerful *chaotic* model.

> **Definition 2.6.6** *An objective W is* finite-memory-determined (over resp. finite, countable, finitely branching, one-player arenas) *if for all (resp. finite, countable, finitely branching, one-player) arenas $\mathcal{A}$, there exist chaotic memory structures $\mathfrak{M}_1$, $\mathfrak{M}_2$ for $\mathcal{A}$ such that $\mathcal{P}_1$ and $\mathcal{P}_2$ each have an optimal strategy respectively based on $\mathfrak{M}_1$ and $\mathfrak{M}_2$ in game $(\mathcal{A}, W)$.*

We then specialize it to chromatic memory structures.

> **Definition 2.6.7** *An objective W is* chromatic-finite-memory-determined (over resp. finite, countable, finitely branching, one-player arenas) *if for all (resp. finite, countable, finitely branching, one-player) arenas $\mathcal{A}$, there exist chromatic memory structures $\mathcal{M}_1$, $\mathcal{M}_2$ such that $\mathcal{P}_1$ and $\mathcal{P}_2$ each have an optimal strategy respectively based on $\mathcal{M}_1$ and $\mathcal{M}_2$ in game $(\mathcal{A}, W)$.*

When in some game, a player has no optimal strategy based on a memory structure (which is finite by definition), we say that this player *needs infinite memory*.

A natural question is: are these two models really distinct? Over finite arenas, finite-memory determinacy and *chromatic* finite-memory determinacy coincide: every chaotic finite-memory structure can be made chromatic up to a blow-up in the size of the memory structure (the blow-up depends on the size of the arena). This was first proved as part of a more general result in [LeR20], with bounds then tightened in [Koz22c].

> **Proposition 2.6.8** ([Koz22c]) *Let $\mathcal{G} = (\mathcal{A}, W)$ be a game played on the finite arena $\mathcal{A} = (V, V_1, V_2, E)$. If $\mathcal{P}_1$ has an optimal strategy in $\mathcal{G}$ based on a chaotic memory structure for $\mathcal{A}$ with n states, then $\mathcal{P}_1$ also has an optimal strategy in $\mathcal{G}$ based on a chromatic memory structure with $(n + 1)^{|V|}$ states.*

Still, we may wonder whether the chaotic memory model brings any succinctness, i.e., whether there is a concrete example with an unavoidable blow-up when going from chaotic to chromatic memory structures. This question was first asked by Kopczyński in his thesis in 2008 [Kop08], and was recently answered positively by Casares [Cas22] (with the gap between both notions then reinforced by Casares, Colcombet, and Lehtinen [CCL22]). The example in [Cas22, Proposition 30] is a family $(W_n)_{n \geq 1}$ of Muller conditions such that in any arena, $\mathcal{P}_1$ always has a chaotic memory structure with two states to play optimally, but needs n states of chromatic memory in some arenas. We will see another original example (from another class of objectives) for which chaotic and chromatic memory requirements do not coincide in Chapter 7. Another example, which we discuss briefly here as it is quite enlightening, originates from the recent preprint [CO22].

[LeR20]: Le Roux (2020), *Time-Aware Uniformization of Winning Strategies*

[Koz22c]: Kozachinskiy (2022), *State Complexity of Chromatic Memory in Infinite-Duration Games*

[Kop08]: Kopczyński (2008), *Half-positional Determinacy of Infinite Games*

[Cas22]: Casares (2022), *On the Minimisation of Transition-Based Rabin Automata and the Chromatic Memory Requirements of Muller Conditions*

[CCL22]: Casares et al. (2022), *On the Size of Good-For-Games Rabin Automata and Its Link with the Memory in Muller Games*

[CO22]: Casares et al. (2022), *Characterising memory in infinite games*
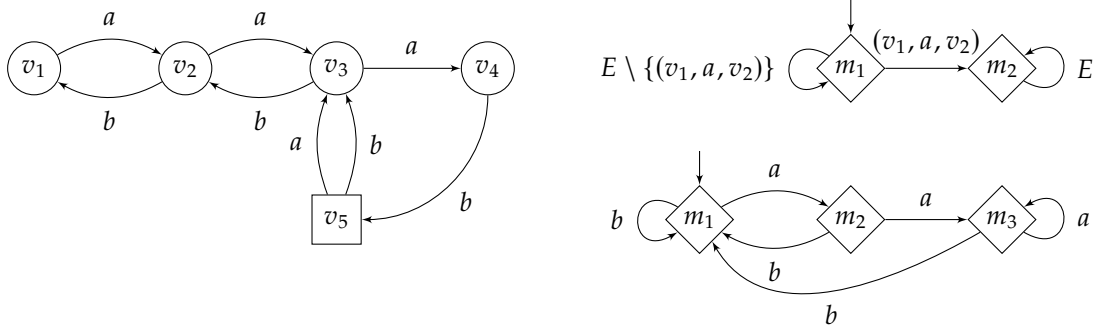
Figure 2.4: Arena in which $\mathcal{P}_1$ has an obvious strategy to play optimally with a chaotic memory structure with two states (depicted at the top right), but playing optimally with two states of chromatic memory is not possible. The chromatic structure with three states at the bottom right suffices to play optimally.

Example 2.6.9 We give a simple example to highlight the difference between the succinctness of chromatic and chaotic structures. We only attempt to give some intuition about the difference between the two models on some chosen arena; complete proofs applying to all arenas can be found in [CO22].

Let $C = \{a, b\}$. We consider the objective $W = C^*a^3C^\omega$, consisting of the infinite words seeing three times the color a in a row at some point. For this objective, $\mathcal{P}_1$ needs some memory to play optimally. Consider the arena in Figure 2.4 (left): the only way to see three a's in a row is to go to v_1, and then move right up to v_4. This requires at least two states of memory, as if the game starts in v_2, v_3, v_4, or v_5, $\mathcal{P}_1$ needs to first go to v_1 before going to v_4, which means that the same edge cannot always be taken in v_2 and v_3. Implementing such a strategy is easy with a chaotic memory structure with two states: simply, change the memory state once the edge (v_1, a, v_2) is seen. Using a chromatic memory structure, to win from v_2 and v_3, we can simply observe when a has been first seen. However, it is more difficult to win from v_4 and v_5; we can actually prove (by exhaustion) that any chromatic structure with two states is insufficient to play optimally. The idea is that the choice of $\mathcal{P}_2$ in v_5 can always be used to fool a chromatic structure with two states. It is however possible to play optimally with a chromatic structure with three states that counts the number of consecutive a's between 0 and 2. It turns out that for objective W, in any arena, $\mathcal{P}_1$ always has an optimal strategy with just two states of chaotic memory, but in some arenas, $\mathcal{P}_1$ requires three states of chromatic memory to play optimally. We will see algorithms that compute automatically minimal chromatic memory structures for this kind of objective in Chapter 7.

[CO22]: Casares et al. (2022), *Characterising memory in infinite games*

We make some light usage of *(ω-)regular expressions* to describe languages of finite and infinite words in examples. This is purely for description purposes, which is why we do not define them formally. We use * for the Kleene star, $^+$ for the Kleene plus (so $a^+ = aa^*$), no symbol for concatenation, $+$ for the union, and $^\omega$ for the infinite repetition. The use of a set (such as C) in an expression refers to any character in it. For instance, if $C = \{a, b\}$, aC denotes $a(a + b)$, i.e., the set $\{aa, ab\}$.

There is therefore a difference in succinctness between chromatic and chaotic structures; chaotic memory structures may sometimes be chosen with fewer states. However, this comes at the cost of needing to specialize the transition function of the memory structure for every arena — it does not permit reasoning about an objective without instantiating an arena. If we additionally look at the number of transitions in the memory

structure, observe that the number of transitions of the chromatic model depends only on the number of memory states and the number of colors. In contrast, the number of transitions of the chaotic model depends on the number of memory states and the number of arena edges. Beyond their number of states, their overall succinctness is therefore debatable.

> **Remark 2.6.10** For games played on *infinite* arenas, we are not aware of a proof of equivalence between finite-memory determinacy and chromatic finite-memory determinacy, nor of an example distinguishing both notions.

We now go one step further and define a stronger kind of finite-memory determinacy.

2.6.3 Arena-independent finite memory

For many objectives (including the *ω-regular ones* — see Section 2.7), a memory structure to play optimally can be chosen entirely *independently of the arena*. To put it another way, for some objectives, there is a "uniform" memory structure that suffices to play optimally in all arenas. We call this property *arena-independent finite-memory determinacy*. Notice that this requires *chromatic* memory structures. Indeed, it would not make sense with *chaotic* memory structures; chaotic memory structures need to access the edges of arenas and are intrinsically *arena-dependent*. This notion is also why it is fruitful to base objectives on an "independent" set of colors rather than on vertices or edges of specific arenas.

> **Definition 2.6.11** *An objective W is* arena-independent-finite-memory-determined (over resp. finite, countable, finitely branching, one-player arenas) *if there exist chromatic memory structures* $\mathcal{M}_1$, $\mathcal{M}_2$ *such that for all (resp. finite, countable, finitely branching, one-player) arenas* $\mathcal{A}$, $\mathcal{P}_1$ *and* $\mathcal{P}_2$ *each have an optimal strategy respectively based on* $\mathcal{M}_1$ *and* $\mathcal{M}_2$ *in game* $(\mathcal{A}, W)$.

The main difference with the previous definitions of finite-memory determinacy is the order of quantifiers: the two memory structures must be instantiated before quantifying universally over arenas.

To be more precise about the memory structure and the player, we use the following terminology. Let $W \subseteq C^\omega$ be an objective, $\mathcal{M}$ be a chromatic memory structure, and $\ell \in \{1, 2\}$ be the index of a player. We say that $\mathcal{M}$ *suffices (to play optimally) for* $\mathcal{P}_\ell$ *for W (in resp. finite, countable, finitely branching, one-player arenas)* if for all (resp. finite, countable, finitely branching, one-player) arenas $\mathcal{A}$, $\mathcal{P}_\ell$ has an optimal strategy based on $\mathcal{M}$ in game $(\mathcal{A}, W)$. Using Lemma 2.5.5, we have that all these determinacy properties are stable by product with arbitrary memory structures: e.g., if $\mathcal{M}$ suffices for a player for W, then for all structures $\mathcal{M}'$, $\mathcal{M} \otimes \mathcal{M}'$ also suffices for this player for W. In particular, up to taking the product
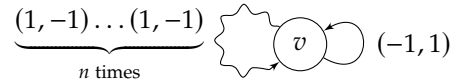
Figure 2.5: Arena in which $\mathcal{P}_1$ needs an amount of memory depending on the parameter $n \geq 1$ to win for a conjunction of two energy objectives. We use a squiggly arrow to indicate a sequence of edges such that taking all the edges in the sequence induces the finite word of the label. Here, this notation therefore conceals n edges (all labeled with $(1, -1)$) and $n - 1$ vertices.

$\mathcal{M} = \mathcal{M}_1 \otimes \mathcal{M}_2$ of the memory structures of each player, observe that an objective W is arena-independent-finite-memory-determined if and only if there exists a memory structure $\mathcal{M}$ that suffices both for $\mathcal{P}_1$ and $\mathcal{P}_2$ for W.

Observe that *memoryless determinacy* is a special case of arena-independent finite-memory determinacy, where the memory structure $\mathcal{M}$ in the definition of arena-independent finite-memory determinacy can be taken to be $\mathcal{M}_{\text{triv}}$.

> **Remark 2.6.12** Unlike finite-memory strategies, the memory model (chromatic or chaotic) has no influence as far as memoryless strategies are concerned: it does not matter whether strategies observe colors or edges, as they cannot distinguish histories from one another anyway. Therefore, results about memoryless determinacy or half-positionality (e.g., in Chapter 8) are independent of the chromatic/chaotic discussion.

Over finite arenas, arena-independent finite-memory determinacy is a stronger notion than finite-memory (or even chromatic finite-memory) determinacy. We illustrate it with a *multi-energy objective*, studied in [VCD+15; CRR14; JLS15].

[VCD+15]: Velner et al. (2015), *The complexity of multi-mean-payoff and multi-energy games* [CRR14]: Chatterjee et al. (2014), *Strategy synthesis for multi-dimensional quantitative objectives* [JLS15]: Jurdziński et al. (2015), *Fixed-Dimensional Energy Games are in Pseudo-Polynomial Time*

> **Example 2.6.13** An *energy objective* has numbers as colors, and is focused on the behavior of the partial sums of colors. This objective is used to model an amount of resources, and the goal is usually to prevent the amount of resources from becoming negative. An often-used reformulation requires that the limit inferior of the partial sums is not $-\infty$, which means that it is possible to start the game with a sufficiently high amount of resources and never run out of resources.
> We consider here a conjunction of two energy objectives, for which we define two dimensions in the colors. Let $C = \mathbb{Z} \times \mathbb{Z}$. Let
>
> $$W = \{(c_1^{(1)}, c_1^{(2)})(c_2^{(1)}, c_2^{(2)}) \ldots \in C^\omega \mid$$
> $$\forall d \in \{1, 2\}, \liminf_{n \to \infty} \sum_{i=1}^{n} c_i^{(d)} > -\infty\}.$$
>
> In other words, to obtain a winning word, the sequence of the partial sums of colors cannot have a subsequence converging toward $-\infty$, and

this along both dimensions. Note that a single energy objective (along one dimension) is memoryless-determined over finite arenas [CdHS03], but W, which is a conjunction of two energy objectives, is not.

We consider the arena in Figure 2.5, depending on a parameter $n \geq 1$. In this arena, $\mathcal{P}_1$ can win from v using finite memory: for each time $\mathcal{P}_1$ goes to the left and gets a partial sum of colors of $(n, -n)$, $\mathcal{P}_1$ can take n times the edge with color $(-1, 1)$ to compensate. This way, the sum of colors never gets below $-n$ along both dimensions. This strategy can be implemented with $n + 1$ states of (chaotic or chromatic) memory.

Yet, using n memory states or fewer is not sufficient to win, as it would induce an ultimately periodic word whose repeating part consists of at most n elements among $(n, -n)$ and $(-1, 1)$. For such a sequence to be winning, $(n, -n)$ needs to be seen at least once, but then $(-1, 1)$ is seen at most $n - 1$ times, which cannot keep the second dimension above a bounded value. This means that for this family of arenas, $\mathcal{P}_1$ can always win using finite memory, but the size of the memory (and more generally, the memory structure) cannot be fixed prior to fixing the arena.

For this objective, in any finite (even two-player) arena, $\mathcal{P}_1$ has an optimal strategy using finite memory [CRR14], but as we saw, the size of the required memory depends on the arena. On the other hand, $\mathcal{P}_2$ always has memoryless optimal strategies. This implies that W is finite-memory-determined over finite arenas, but not arena-independent-finite-memory-determined over finite arenas.

We now turn to games played on infinite arenas. We argue that requiring the existence of *chromatic* finite memory structures to play optimally for an objective in all *infinite* arenas implies the existence of an *arena-independent* finite memory structure. In other words, in the realm of infinite arenas, given the chromatic requirement, arena-independence is for free. This result is not deep and simply relies on the fact that an arbitrary "union" of infinite arenas still constitutes an infinite arena (which is not true when restricted to finite arenas).

Proposition 2.6.14 *Let $W \subseteq C^\omega$ be an objective. The following are equivalent:*

1. *W is chromatic-finite-memory-determined: for all arenas $\mathcal{A}$, there exists a chromatic memory structure $\mathcal{M}^{\mathcal{A}}$ such that both players have an optimal strategy based on $\mathcal{M}^{\mathcal{A}}$ in $\mathcal{A}$;*
2. *W is arena-independent-finite-memory-determined: there exists a chromatic memory structure $\mathcal{M}$ such that for all arenas $\mathcal{A}$, both players have an optimal strategy based on $\mathcal{M}$ in $\mathcal{A}$.*

Proof. It is clear that 2. $\implies$ 1., as 2. means that there is a (fixed) memory structure that suffices in each arena. We now show 1. $\implies$ 2. We proceed by contraposition. Assume that 2. does not hold, i.e., that for all memory structures $\mathcal{M}$, there exists an arena $\mathcal{A}^{\mathcal{M}} = (V^{\mathcal{M}}, V_1^{\mathcal{M}}, V_2^{\mathcal{M}}, E^{\mathcal{M}})$ such that
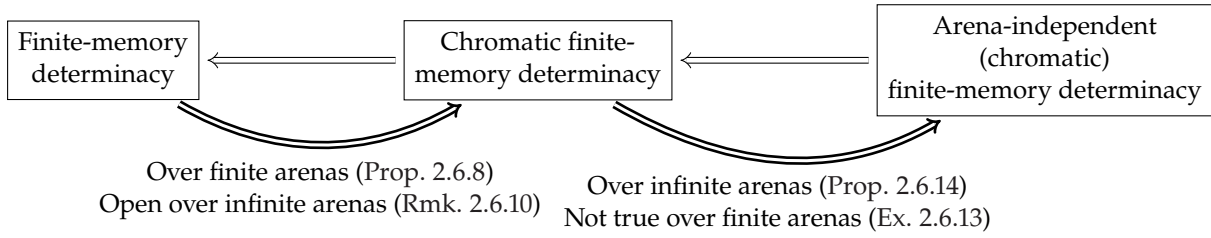
Figure 2.6: Relations between types of finite-memory determinacy. Thin implications hold by definition. Thick implications only hold in particular contexts.

at least one player does not have an optimal strategy based on $\mathcal{M}$ in $\mathcal{A}^{\mathcal{M}}$. We consider the arena $\mathcal{A} = (\uplus_{\mathcal{M}} V^{\mathcal{M}}, \uplus_{\mathcal{M}} V_1^{\mathcal{M}}, \uplus_{\mathcal{M}} V_2^{\mathcal{M}}, \uplus_{\mathcal{M}} E^{\mathcal{M}})$ consisting in the "disjoint union" over all memory structures $\mathcal{M}$ of the arenas $\mathcal{A}^{\mathcal{M}}$. Clearly, no strategy based on a chromatic memory structure suffices to play optimally in $\mathcal{A}$; this shows that 1. does not hold. $\square$

2.6.4 Overview

We sum up the relations between the three kinds of finite-memory determinacy in Figure 2.6.

When considering finite arenas, we have discussed that finite-memory determinacy and chromatic finite-memory determinacy correspond (up to a blow-up in the size of the memory — Proposition 2.6.8), and we have distinguished them from arena-independent finite-memory determinacy with an example (Example 2.6.13). When considering infinite arenas, we have that chromatic finite-memory and arena-independent finite-memory determinacy correspond (Proposition 2.6.14), but we do not know whether they can be distinguished from finite-memory determinacy with an example (Remark 2.6.10). We leave as an open question the existence of an objective that is finite-memory-determined over infinite arenas, but not chromatic-finite-memory-determined.

2.7 Automata and ω-regular objectives

Automata. We will often specify objectives using *automata*, and objectives that can be defined using automata will be central for finite-memory determinacy. We start by defining a common base for all kinds of automata that we consider in this work.

Definition 2.7.1 *A* (deterministic) automaton structure *is a tuple* $\mathcal{S} = (Q, \Sigma, q_{\mathsf{init}}, \delta)$ *where Q is a finite set of* states, *Σ is a non-empty* alphabet *(often $\Sigma = C$ is the set of colors), $q_{\mathsf{init}} \in Q$ is an* initial state, *and $\delta \colon Q \times \Sigma \to Q$ is a (deterministic, complete)* update function.

Except for a few clearly labeled occurrences of non-deterministic finite automata in Chapter 4, all automata in this work are deterministic. We sometimes omit the word *deterministic* when there is no ambiguity.

We write $\delta^*\colon Q \times \Sigma^* \to Q$ for the natural extension of δ to sequences of colors (defined exactly like α^*_{upd} for memory structures). An element of $Q \times \Sigma$ is called a *transition* of an automaton structure (as we consider deterministic structures, a state/letter pair determines unequivocally what state comes next). For $q_1, q_2 \in Q$, we write $L^{\mathcal{S}}_{q_1,q_2}$ for the language of words $w \in \Sigma^*$ such that $\delta^*(q_1, w) = q_2$. We drop the superscript $\mathcal{S}$ if the automaton structure considered is clear in the context. Given an infinite word $w = c_1 c_2 \ldots \in \Sigma^\omega$, the *infinite run of $\mathcal{S}$ on w* is the sequence $\varrho = (q_0, c_1)(q_1, c_2) \ldots \in (Q \times \Sigma)^\omega$ where $q_0 = q_{\mathsf{init}}$, and for $i \geq 0$, $\delta(q_i, c_{i+1}) = q_{i+1}$. If $w = c_1 \ldots c_n \in \Sigma^*$ is finite, the *finite run of $\mathcal{S}$ on w* is the sequence $\varrho = (q_0, c_1) \ldots (q_{n-1}, c_n) \in (Q \times \Sigma)^*$ such that $q_0 = q_{\mathsf{init}}$ and for $0 \leq i < n - 1$, $\delta(q_i, c_{i+1}) = q_{i+1}$.

The word *language* simply refers to a set of words.

Automaton structures $\mathcal{S} = (Q, \Sigma, q_{\mathsf{init}}, \delta)$ are syntactically almost the same as memory structures $\mathcal{M} = (M, m_{\mathsf{init}}, \alpha_{\mathsf{upd}})$ (except that the alphabet for memory structures is always the set of colors C and is implicit). We distinguish the two notions as they will be used for different purposes; it is convenient to use different symbols not to confuse both objects as there will be examples relying both on an automaton structure and a memory structure. In Chapter 5, we will actually (slightly abusively) use memory structures as automaton structures.

We now define three classical kinds of automata, all building on the definition of an automaton structure. As such, they inherit notations related to automaton structures such as transitions, languages $L^{\mathcal{S}}_{q_1,q_2}$, runs. . . The three kinds are *deterministic finite automata*, *deterministic Büchi automata*, and *deterministic parity automata*.

> **Definition 2.7.2** *A deterministic finite automaton (abbreviated* DFA*) is a tuple $\mathcal{D} = (Q, \Sigma, q_{\mathsf{init}}, \delta, F)$ where $(Q, \Sigma, q_{\mathsf{init}}, \delta)$ is an automaton structure and $F \subseteq Q$ is a set of* final states.

The *language recognized by DFA $\mathcal{D} = (Q, \Sigma, q_{\mathsf{init}}, \delta, F)$*, denoted $\mathcal{L}(\mathcal{D})$, is the set of finite words $w \in \Sigma^*$ such that $\delta^*(q_{\mathsf{init}}, w) \in F$. DFAs (defined with a finite alphabet Σ) recognize exactly the regular languages.

> **Definition 2.7.3** *A* (transition-based) *deterministic Büchi automaton (abbreviated* DBA*) is a tuple $\mathcal{B} = (Q, \Sigma, q_{\mathsf{init}}, \delta, B)$ where $(Q, \Sigma, q_{\mathsf{init}}, \delta)$ is an automaton structure and $B \subseteq Q \times \Sigma$ is a set of* Büchi transitions.

The *language recognized by DBA $\mathcal{B} = (Q, \Sigma, q_{\mathsf{init}}, \delta, B)$*, denoted $\mathcal{L}(\mathcal{B})$, is the set of infinite words $w \in \Sigma^\omega$ such that the infinite run of $\mathcal{B}$ on w visits infinitely many times a Büchi transition in B.

> **Definition 2.7.4** *A* (transition-based) *deterministic parity automaton (abbreviated* DPA*) is a tuple $\mathcal{P} = (Q, \Sigma, q_{\mathsf{init}}, \delta, p)$ where $(Q, \Sigma, q_{\mathsf{init}}, \delta)$ is an automaton structure and, for some $n \in \mathbb{N}$, $p\colon Q \times \Sigma \to \{0, \ldots, n\}$ is a* priority function *assigning priorities to transitions of $\mathcal{P}$.*

The *language recognized by DPA* $\mathcal{P} = (Q, \Sigma, q_{\text{init}}, \delta, p)$, denoted $\mathcal{L}(\mathcal{P})$, is the set of infinite words $w \in \Sigma^\omega$ such that, if $(q_0, c_1)(q_1, c_2)\ldots$ is the infinite run of $\mathcal{P}$ on w, then $\limsup_i p(q_i, c_{i+1})$ is even. This last property can be stated as "the maximal priority seen infinitely often is even".

> **Remark 2.7.5** For Büchi and parity automata, we consider *transition-based* acceptance conditions, because we define the set B and the function p on the *transitions*. Traditionally, acceptance conditions were defined on the states. Transition-based and state-based automata are expressively equivalent: a state-based automaton can be converted into a transition-based automaton of the same size recognizing the same language, and a transition-based automaton can be converted into a polynomially larger state-based automaton, where the blow-up depends on the acceptance condition. Transition-based automata are therefore more succinct with respect to the number of states in general.
>
> Transition-based automata are becoming increasingly frequent in the literature since they present advantages for multiple problems (e.g., to build smaller Büchi automata [GL02], to build memory structures for Muller conditions [Cas22], to minimize and define canonical automata [AK22]). This is also how automata are encoded in recent software tools (see, e.g., Spot [DLF+16] and Owl [KMS18]). In our work, it turns out that in multiple occurrences (discussed in due time), the transition-based automata allowed for more elegant statements, so we opted for this convention.

We have defined automaton structures to be *complete*, i.e., every color defines a transition from every state. This implies that all deterministic automata in this thesis are complete, which is a technical convenience.

We sometimes build automata (DFAs, DBAs, or DPAs) starting from an automaton structure. In such a case, we say that a DFA, DBA, or DPA is *built on top of the structure*.

> **Definition 2.7.6** *A DFA* $\mathcal{D} = (Q, \Sigma, q_{\text{init}}, \delta, F)$ *(or a DBA* $\mathcal{B} = (Q, \Sigma, q_{\text{init}}, \delta, B)$, *a DPA* $\mathcal{P} = (Q, \Sigma, q_{\text{init}}, \delta, p)$*) is* built on top of automaton structure $\mathcal{S}$ *if* $\mathcal{S} = (Q, \Sigma, q_{\text{init}}, \delta)$.

> **Remark 2.7.7** Automata are usually defined on a finite alphabet, whereas our alphabet is often the set of colors C and can have any non-zero cardinality. However, given a DFA (resp. DBA, DPA) built on top of an automaton structure $\mathcal{S} = (Q, \Sigma, q_{\text{init}}, \delta)$, as there are finitely many states in $\mathcal{S}$ and finitely many "levels of acceptance", there are in practice only finitely many "truly different" equivalence classes of letters in the alphabet. For a DFA $(\mathcal{S}, F)$ (resp. a DBA $(\mathcal{S}, B)$, a DPA $(\mathcal{S}, p)$), two letters $c_1, c_2 \in \Sigma$ can be assumed to be equivalent if for all $q \in Q$, $\delta(q, c_1) = \delta(q, c_2)$ (resp. $\delta(q, c_1) = \delta(q, c_2)$ and $(q, c_1) \in B$ if and only if $(q, c_2) \in B$, $\delta(q, c_1) = \delta(q, c_2)$ and $p(q, c_1) = p(q, c_2)$).

[GL02]: Giannakopoulou et al. (2002), *From States to Transitions: Improving Translation of LTL Formulae to Büchi Automata*

[Cas22]: Casares (2022), *On the Minimisation of Transition-Based Rabin Automata and the Chromatic Memory Requirements of Muller Conditions*

[AK22]: Abu Radi et al. (2022), *Minimization and Canonization of GFG Transition-Based Automata*

[DLF+16]: Duret-Lutz et al. (2016), *Spot 2.0 — A Framework for LTL and ω-Automata Manipulation*

[KMS18]: Křetínský et al. (2018), *Owl: A Library for ω-Words, Automata, and LTL*
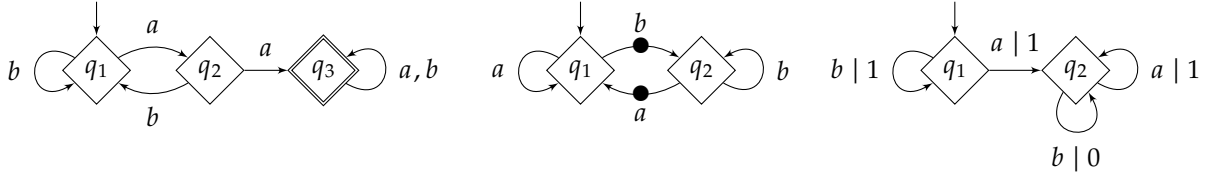
Figure 2.7: Three kinds of deterministic automata considered in this thesis. We use diamonds to represent automaton states (as for memory states). The initial state is indicated with an ingoing arrow. Final states (in F) of DFAs are represented with a double border. Büchi transitions (in B) of Büchi automata are decorated with a bullet •. A transition of a parity automaton from q to q' with label $c \mid k$ means that $\delta(q,c) = q'$ and $p(q,c) = k$.

We illustrate each of the three kinds of automata with an example.

> **Example 2.7.8** We depict an example of each kind of automaton in Figure 2.7. In every case, the alphabet is $C = \{a, b\}$.
>
> ▶ The DFA $\mathcal{D}$ on the left accepts the finite words that reach the accepting state q_3 (drawn with a double border) at some point. To reach q_3, a word must see twice the color a in a row. A regular expression for the language recognized by $\mathcal{D}$ (i.e., $\mathcal{L}(\mathcal{D})$) is $C^* a a C^*$.
> ▶ The DBA $\mathcal{B}$ in the center accepts the infinite words that see infinitely often a transition in B, i.e., depicted with a • symbol. Here, it means that it accepts all the infinite words that do not end with a^ω or with b^ω. In other words, the language recognized by this DBA is $\mathcal{L}(\mathcal{B}) = \mathsf{Büchi}(a) \cap \mathsf{Büchi}(b)$.
> ▶ The DPA $\mathcal{P}$ on the right accepts the infinite words whose maximal priority seen infinitely often is even. The priorities are written on transitions, next to the letter used for transitions. Here, the only way for an infinite word to be accepted is to see only the priority 0 from some point on. To do so, the word must see at least one but finitely many times the letter a. An ω-regular expression for $\mathcal{L}(\mathcal{P})$ is $b^* a C^* b^\omega$.

DBAs and DPAs are related: any language recognized by a DBA can also be recognized by a DBA, but the converse does not hold.

> **Remark 2.7.9** A DBA is a special kind of DPA: for a DBA $\mathcal{B} = (Q, \Sigma, q_{\mathsf{init}}, \delta, B)$, the DPA $\mathcal{P} = (Q, \Sigma, q_{\mathsf{init}}, \delta, p)$ where for $(q,c) \in Q \times \Sigma$,
>
> $$p(q,c) = \begin{cases} 2 & \text{if } (q,c) \in B \\ 1 & \text{if } (q,c) \notin B \end{cases}$$
>
> recognizes the same set of infinite words (infinitely many transitions with priority 2, that is, transitions in B, have to be seen to accept a run). Therefore, DPAs are at least as expressive as DBAs.
> Moreover, DBAs only recognize a proper subset of the languages recognized by DPAs. In other words, not every DPA can be converted into a DBA recognizing the same language [Wag79]. For example, $\mathcal{L}(\mathcal{P})$ in Example 2.7.8 is not recognizable by a DBA (a close proof can be found

[Wag79]: Wagner (1979), *On ω-Regular Sets*

in [BK08, Theorem 4.50]).

ω-regularity. Sets of infinite words recognizable by a DPA are usually called *ω-regular*. There are multiple equivalent definitions of ω-regular sets (they are the ones definable by *ω-regular expressions*, *non-deterministic Büchi automata*, *deterministic Muller automata*. . .), but a convenient definition for our purposes is the one using DPAs.

> **Definition 2.7.10** *A subset of Σ^ω is ω-regular if it is recognizable by a deterministic parity automaton.*

It follows from Remark 2.7.9 that the languages recognized by DBAs are ω-regular, but that DBAs are not expressive enough to recognize all ω-regular languages. The class of ω-regular languages is closed under Boolean operations (union, intersection, and complement). Depending on the representation used for ω-regular objectives, proving closeness under some of these operations is more or less complicated. Using the representation as deterministic parity automata, proving that ω-regular languages are closed under complement is easy: given an objective recognized by a DPA $(Q, C, q_{\mathsf{init}}, \delta, p)$, the DPA $(Q, C, q_{\mathsf{init}}, \delta, p')$ where $p' = p + 1$ recognizes the complement of the language. We refer to surveys [GTW02; Bok18] about different automatic ways to represent ω-regular languages and the blow-ups involved when applying Boolean operations to them.

We will naturally consider ω-regular *objectives*, which are sets of infinite words in C^ω recognizable by a DPA. We have already mentioned in Chapter 1 the finite-memory determinacy of ω-regular objectives, which appears (with varying degree of explicitness) in [BL69; Rab69; GH82; McN93]. A thorough account using a very close formalism (and considering explicitly games played on *infinite* graphs) is also available in [Zie98]. From these works, in our vocabulary, we can even deduce the *arena-independent* finite-memory determinacy of ω-regular objectives.

> **Theorem 2.7.11** ([GH82; McN93]) *The ω-regular objectives are arena-independent-finite-memory-determined.*

Sketch of proof. Multiple arguments lead to this result, through the various representations of ω-regular languages. Initial proofs [GH82; McN93] used the representation of ω-regular objectives as deterministic Muller automata and built the (arena-independent) *later appearance record (LAR)* memory structure.

For our purposes, we highlight the following (slightly informal) argument, which will be revisited in Chapter 5. Every ω-regular objective admits a representation as a deterministic parity automaton. We take the underlying structure of this automaton as a memory structure. With this additional information, we effectively reduce every game into a larger

game with a parity condition (by considering the *product of the arena and the memory structure*, which is formalized in Chapter 4). As parity conditions are memoryless-determined (Theorem 2.6.3), there is no need for extra information to build optimal strategies; the information from the memory structure suffices. □

Finally, we restate a well-known lemma about ω-regular objectives: if two ω-regular objectives are not equal, then they are distinguished by an *ultimately periodic* word (i.e., a word that can be written as $w_1(w_2)^\omega$ for some $w_1 \in C^*$ and $w_2 \in C^+$). Ultimately periodic words can easily be finitely represented, and this lemma will be used to force some behaviors to appear in *finite* game arenas.

> **Lemma 2.7.12** *Every non-empty ω-regular language contains an ultimately periodic word. In particular, let $W_1, W_2 \subseteq C^\omega$ be two ω-regular objectives. If $W_1 \neq W_2$, then there exist $w_1 \in C^*$ and $w_2 \in C^+$ such that $w_1(w_2)^\omega \in W_1 \setminus W_2$ or $w_1(w_2)^\omega \in W_2 \setminus W_1$.*

Sketch of proof. The first claim is standard and follows from McNaughton's theorem [McN66]: a non-empty ω-regular language admits a representation through an automaton accepting at least one word. Due to the acceptance condition (e.g., Büchi or parity), this implies there is at least one reachable "accepting cycle", i.e., a cycle of the automaton that, when repeated, induces an accepted word. Therefore, there is an ultimately periodic run leading to that cycle and then looping around it ad infinitum, which induces an ultimately periodic word accepted by the automaton.

[McN66]: McNaughton (1966), *Testing and Generating Infinite Sequences by a Finite Automaton*

For the second claim, if $W_1 \neq W_2$, then $W_1 \setminus W_2 \neq \emptyset$ or $W_2 \setminus W_1 \neq \emptyset$. Without loss of generality, we assume that $W_1 \setminus W_2 \neq \emptyset$. Objective $W_1 \setminus W_2$ is ω-regular (as ω-regular objectives are closed under complement and intersection) and non-empty. By the first claim, there is an ultimately periodic word in $W_1 \setminus W_2$. □

The contrapositive of the previous lemma yields that two ω-regular languages are equal if and only if they coincide on the ultimately periodic words.

2.8 Continuations and congruences

It is often useful to compare different situations that players may find themselves in; besides the current vertex of the arena, a situation is determined by what already happened, and what already happened determines what must be seen to win in the future.

Definition 2.8.1 *For an objective $W \subseteq C^\omega$ and a finite word $w \in C^*$, the* winning continuations of w *consist in the set denoted $w^{-1}W$ and defined as*

$$\{w' \in C^\omega \mid ww' \in W\}.$$

A set of winning continuations is sometimes called a *left quotient of W* or *residual language of W* in the literature.

Observe that for all objectives W, it holds that $\varepsilon^{-1}W = W$. We can compare pairs of finite words using their sets of winning continuations.

Definition 2.8.2 *Let $W \subseteq C^\omega$ be an objective. The* prefix preorder of W *is the relation $\preceq_W \subseteq C^* \times C^*$ defined by $w_1 \preceq_W w_2$ if $w_1^{-1}W \subseteq w_2^{-1}W$ — we simply order the sets of winning continuations by inclusion.*

The prefix preorder is in general a preorder but not an order, since two distinct finite words may have the same sets of winning continuations. It is also partial in general, as two finite words may have incomparable sets of winning continuations.

Definition 2.8.3 *Let $W \subseteq C^\omega$ be an objective. The* right congruence of W *is the relation $\sim_W \subseteq C^* \times C^*$ defined by $w_1 \sim_W w_2$ if $w_1^{-1}W = w_2^{-1}W$.*

Notice that $\sim_W = \preceq_W \cap \succeq_W$.

The right congruence of an objective is an equivalence relation. We denote $\prec_W = \preceq_W \setminus \sim_W$ for the *strict prefix preorder of W*. We drop the subscripts W when there is no ambiguity on the objective being considered.

Example 2.8.4 Let $C = \{a, b\}$. We consider objective $\mathcal{L}(\mathcal{B}) = \mathsf{Büchi}(a) \cap \mathsf{Büchi}(b)$ from Example 2.7.8. For this objective, we have that all finite words have the same winning continuations: for all finite words $w \in C^*$, $w^{-1}\mathcal{L}(\mathcal{B}) = \mathcal{L}(\mathcal{B})$. Therefore, all finite words are equivalent for $\sim_{\mathcal{L}(\mathcal{B})}$; the right congruence has a single equivalence class.
We now consider objective $\mathcal{L}(\mathcal{P}) = b^*aC^*b^\omega$ from Example 2.7.8. Here, the right congruence $\sim_{\mathcal{L}(\mathcal{P})}$ has two equivalence classes: finite words that have not seen a yet and finite words that have seen a. A representative of these classes is given respectively by ε and a. These two words are comparable for $\preceq_{\mathcal{L}(\mathcal{P})}$; any continuation winning after ε is also winning after a, so $\varepsilon \preceq_{\mathcal{L}(\mathcal{P})} a$. However, we do not have $a \preceq_{\mathcal{L}(\mathcal{P})} \varepsilon$, as $b^\omega \in a^{-1}\mathcal{L}(\mathcal{P}) \setminus \varepsilon^{-1}\mathcal{L}(\mathcal{P})$. Hence, we have $\varepsilon \prec_{\mathcal{L}(\mathcal{P})} a$.

The prefix preorder and the right congruence of an objective are *algebraic (pre)congruences*: the algebraic operation of reading colors preserves these relations.

Lemma 2.8.5 *Let $W \subseteq C^\omega$ be an objective, $\preceq$ be its prefix preorder, and $w_1, w_2 \in C^*$ be finite words. If $w_1 \preceq w_2$, then for all $w \in C^*$, $w_1w \preceq w_2w$. In particular, if $w_1 \sim w_2$ (resp. $w_1 \prec w_2$), then for all $w \in C^*$, $w_1w \sim w_2w$ (resp. $w_1w \preceq w_2w$).*

Proof. We assume that $w_1 \preceq w_2$, i.e., that $w_1^{-1}W \subseteq w_2^{-1}W$. Let $w \in C^*$. We have that

$$
\begin{aligned}
(w_1 w)^{-1} W &= \{w' \in C^\omega \mid w_1 w w' \in W\} \\
&= \{w' \in C^\omega \mid w w' \in w_1^{-1} W\} \\
&\subseteq \{w' \in C^\omega \mid w w' \in w_2^{-1} W\} \quad \text{as } w_1^{-1} W \subseteq w_2^{-1} W \\
&= \{w' \in C^\omega \mid w_2 w w' \in W\} \\
&= (w_2 w)^{-1} W.
\end{aligned}
$$

Hence, $w_1 w \preceq w_2 w$.

The other properties follow from the first one as $\sim \; = \; \preceq \cap \succeq$ and $\prec \; \subseteq \; \preceq$. □

For many classical objectives, all finite words have the same winning continuations. This is for instance the case of objectives $\mathsf{Büchi}(a)$, $\mathsf{Parity}(n)$, $\mathsf{Muller}(\mathcal{F})$, $\mathsf{MP}^{\geq 0}$, and $\mathsf{Büchi}(a) \cap \mathsf{Büchi}(b)$; what matters for these objectives is the behavior "at infinity", and whether a word is winning or not is not impacted by finite prefixes of the word. Such objectives are called *prefix-independent*. On the other hand, objectives $\mathsf{Reach}(a)$ and $b^* a C^* b^\omega$ are not prefix-independent (in both cases, finite words ε and a do not have the same winning continuations).

> **Definition 2.8.6** *An objective $W \subseteq C^\omega$ is* prefix-independent *if its right congruence has a single equivalence class. Equivalently, W is prefix-independent if for all $w \in C^*$ and $w' \in C^\omega$, $w' \in W$ if and only if $ww' \in W$.*

When an ω-regular objective is specified with a DBA or a DPA, it is convenient to extend the prefix preorder and the right congruence to the automaton states. This is well-defined thanks to the following elementary lemma.

> **Lemma 2.8.7** *Let $\mathcal{P} = (Q, C, q_{\mathsf{init}}, \delta, p)$ be a deterministic parity automaton, and $W \subseteq C^\omega$ be the ω-regular objective it recognizes. For $w, w' \in C^*$, if $\delta^*(q_{\mathsf{init}}, w) = \delta^*(q_{\mathsf{init}}, w')$, then $w \sim_W w'$.*

Note that this lemma and the following notations also apply to deterministic Büchi automata, as they are a special kind of deterministic parity automata (Remark 2.7.9).

Proof. Let $w, w' \in C^*$. If w and w' reach the same state q of the automaton, then by construction, their winning continuations are exactly the words in the language recognized by DPA (Q, C, q, δ, p). Hence, w and w' have the same winning continuations; in other words, $w \sim_W w'$. □

This means that all words reaching the same automaton state are in the same equivalence class for $\sim$: formally, for all states $q \in Q$, $L_{q_{\mathsf{init}}, q} \subseteq [w]_\sim$ for any $w \in L_{q_{\mathsf{init}}, q}$. We can therefore extend the idea of winning continuations to automaton states; for $\mathcal{P} = (Q, C, q_{\mathsf{init}}, \delta, p)$, $W \subseteq C^\omega$ the objective it recognizes, and $q \in Q$, we write

$$
q^{-1} W = w^{-1} W
$$

for some word $w \in L_{q_{\text{init}},q}$. Equivalently, $q^{-1}W$ is the ω-regular language recognized by DPA (Q, C, q, δ, p) (the same as $\mathcal{P}$ but with q as an initial state).

In particular, the number of equivalence classes for the right congruence of W is upper bounded by the number of elements in Q.

Lemma 2.8.8 *Let $W \subseteq C^\omega$ be an ω-regular objective. The right congruence $\sim$ of W has a finite index. More precisely, if W is recognized by a DPA $\mathcal{P} = (Q, C, q_{\text{init}}, \delta, p)$, then the right congruence of W has index at most $|Q|$.*

We recall that the index of an equivalence relation is its number of equivalence classes.

Proof. As it is ω-regular, objective W is recognized by some DPA $\mathcal{P} = (Q, C, q_{\text{init}}, \delta, p)$. The family $(L_{q_{\text{init}},q})_{q \in Q}$ is a finite partition of C^* (we recall that our automata are deterministic and complete). As every set $L_{q_{\text{init}},q}$ is a subset of some equivalence class for the right congruence $\sim$ of W, this right congruence has at most $|Q|$ equivalence classes. $\square$

Some objectives have a representation using an automaton with exactly one state per equivalence class of the right congruence. However, in some cases, even a minimal automaton (w.r.t. the number of states) recognizing an objective may require more states than the number of equivalence classes. This is for instance the case of objective $\text{Büchi}(a) \cap \text{Büchi}(b)$ from Example 2.7.8: it is prefix-independent and its right congruence has a single equivalence class, but any DBA (or even DPA) recognizing it needs at least two states. In general, $(L_{q_{\text{init}},q})_{q \in Q}$ is therefore a finer partition of C^* than $C^*/_\sim$.

Remark 2.8.9 As for the simpler case of *regular* languages of *finite* words, the above lemma states that ω-regularity of an objective (a language of infinite words) implies that its right congruence has a finite index. For regular languages of finite words, the converse is true; this is the central Myhill-Nerode theorem [Ner58]. However, for languages of infinite words, having a finite index is not sufficient to guarantee ω-regularity. For instance, objective $\text{MP}^{\geq 0}$ is prefix-independent and its right congruence has therefore index 1, but $\text{MP}^{\geq 0}$ is not ω-regular.

[Ner58]: Nerode (1958), *Linear Automaton Transformations*

We extend the prefix preorder and the right congruence to automaton states: we define $\preceq_\mathcal{P} \subseteq Q \times Q$ and $\sim_\mathcal{P} \subseteq Q \times Q$ for the relations such that $q_1 \preceq_\mathcal{P} q_2$ if $q_1^{-1}W \subseteq q_2^{-1}W$, and $q_1 \sim_\mathcal{P} q_2$ if $q_1^{-1}W = q_2^{-1}W$. We also frequently drop the subscript $\mathcal{P}$ when there is no ambiguity on the automaton being considered. These relations are still (pre)congruences.

Lemma 2.8.10 *Let $\mathcal{P} = (Q, C, q_{\text{init}}, \delta, p)$ be a deterministic parity automaton, and W be the ω-regular objective it recognizes. Then,*

▶ *for $q_1, q_2 \in Q$, if $q_1 \preceq_\mathcal{P} q_2$, then for all $w \in C^*$, $\delta^*(q_1, w) \preceq_\mathcal{P} \delta^*(q_2, w)$;*
▶ *for $q_1, q_2 \in Q$, if $q_1 \sim_\mathcal{P} q_2$, then for all $w \in C^*$, $\delta^*(q_1, w) \sim_\mathcal{P} \delta^*(q_2, w)$.*
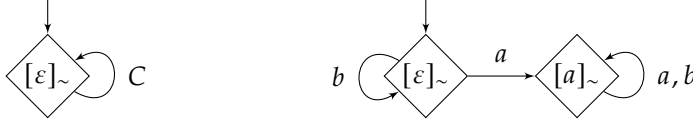
Proof. For the first item, let $q_1, q_2 \in Q$ be such that $q_1 \preceq_\mathcal{P} q_2$ and $w \in C^*$. Let $q_1' = \delta^*(q_1, w)$ and $q_2' = \delta^*(q_2, w)$. We show that $q_1' \preceq_\mathcal{P} q_2'$, i.e., that $(q_1')^{-1}W \subseteq (q_2')^{-1}W$. Let $w' \in (q_1')^{-1}W$. This implies that $ww' \in q_1^{-1}W$. As $q_1 \preceq_\mathcal{P} q_2$, we also have that $ww' \in q_2^{-1}W$. This implies that $w' \in (q_2')^{-1}W$.

For the second item, we simply use the first item and the fact that $\sim_\mathcal{P} = \preceq_\mathcal{P} \cap \succeq_\mathcal{P}$. □

For the case of regular languages of finite words, the way the equivalence classes of the right congruence are structured directly gives a minimal deterministic finite automaton *recognizing* the language (this is the Myhill-Nerode theorem [Ner58]). The situation is once again more complex for languages of infinite words: when the right congruence has a finite index, there is still an automaton that "classifies" the finite words [Sta83; MS97], but it may not be complex enough to actually *recognize* the language of infinite words with a reasonable acceptance condition (e.g., Büchi or parity). This object, which we call *prefix classifier*, may still bring insight into a language of infinite words. In Chapter 5, we will prove that even if insufficient alone, it is a useful block to build automata recognizing ω-regular languages.

[Ner58]: Nerode (1958), *Linear Automaton Transformations*

[Sta83]: Staiger (1983), *Finite-State ω-Languages*
[MS97]: Maler et al. (1997), *On Syntactic Congruences for Omega-Languages*

> **Definition 2.8.11** *Let $W \subseteq C^\omega$ be an objective whose right congruence $\sim$ has a finite index. We associate a natural automaton structure $\mathcal{S}_W = (Q_W, C, q_{init}^W, \delta_W)$ with W such that Q_W is the set of equivalence classes of $\sim$, $q_{init}^W = [\varepsilon]_\sim$, and $\delta_W([w]_\sim, c) = [wc]_\sim$. We call structure $\mathcal{S}_W$ the* prefix classifier *of W.*

The prefix classifier has been called *minimal-state automaton* in the literature [MS97]. For the purposes of this thesis (often, finding memory structures with minimally many states), we believe that the term *prefix classifier* brings less ambiguity.

This transition function of the prefix classifier is well-defined since if $w_1 \sim w_2$, then for all $c \in C$, $w_1c \sim w_2c$ (Lemma 2.8.5). Hence, the choice of representatives for the equivalence classes does not have an impact on this definition. As ω-regular objectives have a right congruence with finite index (Lemma 2.8.8), every ω-regular objective has a well-defined (finite) prefix classifier.

> **Example 2.8.12** We exhibit the prefix classifiers of two objectives from Example 2.7.8, depicted in Figure 2.8.
> The prefix classifier of Büchi(a) ∩ Büchi(b) (and actually, of any prefix-independent objective) has a single state, corresponding to the only equivalence class of the right congruence. Observe that it is not possible to build a DPA on top of this prefix classifier to recognize Büchi(a) ∩ Büchi(b); no matter how priorities are assigned to the two transitions, Büchi(a) ∩ Büchi(b) cannot be recognized by a DPA with just one state.

The prefix classifier of $b^*aC^*b^\omega$ has two states (corresponding to the classes of finite words that have not seen a, and finite words that have seen a). In this case, it actually corresponds to the underlying automaton structure of DPA $\mathcal{P}$ from Example 2.7.8. This means that $b^*aC^*b^\omega$ can be recognized by a DPA built on top of its prefix classifier, unlike $\mathsf{Büchi}(a) \cap \mathsf{Büchi}(b)$. Notice that objectives $\mathsf{Reach}(a)$ and $\mathsf{Safe}(a)$ have the same prefix classifier as $b^*aC^*b^\omega$.

CHARACTERIZING
FINITE MEMORY REQUIREMENTS

From memoryless
to finite-memory determinacy | 3

Our goal in this first part titled *Characterizing finite memory requirements* is to understand the general mechanisms leading various objectives to admit simple optimal strategies. We seek to further the understanding of memory requirements and provide general tools that simplify their study. In this chapter, we highlight existing general results about the memory requirements of various objectives (mostly about memoryless determinacy in various contexts), show some of their limits, and lay the groundwork for new results about finite-memory determinacy.

This chapter serves as an introduction and a motivation to Chapters 4 and 5, which contain our main contributions to these questions.

3.1 Finite game graphs

We first discuss the setting of games played on *finite* graphs. We discuss an elegant characterization of memoryless determinacy in this setting, and then discuss its potential generalization to finite-memory determinacy.

3.1.1 Characterization of memoryless determinacy

Following sufficient conditions for memoryless determinacy [GZ04], the first characterization of objectives that are memoryless-determined over finite graphs was established in [GZ05]. This characterization goes through the conjunction of two conditions of the objectives, called *monotony* and *selectivity*. We give here informal descriptions of these concepts (formal definitions and generalizations will follow in Chapter 4).

Roughly, *monotony* is focused on *prefixes*: it requires that if an ultimately periodic word $x(y_1)^\omega$ is winning and another one $x(y_2)^\omega$ is losing, then their "winning status" cannot be swapped by replacing prefix x, i.e., we cannot have $x'(y_1)^\omega$ losing and $x'(y_2)^\omega$ winning for any $x' \in C^*$. This property is satisfied by prefix-independent objectives, but is more general. For instance, $\mathsf{Reach}(a)$ and $\mathsf{Safe}(a)$ are monotone but not prefix-independent.

Selectivity roughly requires that combining cycles brings no benefit: if it is possible to create a winning infinite word by aggregating finite words from

[GZ04]: Gimbert et al. (2004), *When Can You Play Positionally?*

[GZ05]: Gimbert et al. (2005), *Games Where You Can Play Optimally Without Any Memory*

The results from [GZ05] actually use *preference relations*, a more general framework than qualitative objectives, which we will introduce in Chapter 4.

two distinct sets, then it is also possible to create a winning word by taking finite words from just one of these sets. Similar-looking notions (*fairly mixing*, *concave*, *submixing* [GZ04; Kop06; GK14]) had been defined in other attempts in the literature, but they slightly differ and are incomparable to selectivity.

[GZ04]: Gimbert et al. (2004), *When Can You Play Positionally?*
[Kop06]: Kopczyński (2006), *Half-Positional Determinacy of Infinite Games*
[GK14]: Gimbert et al. (2014), *Submixing and Shift-Invariant Stochastic Games*

Perhaps more elegant and handy is the following characterization of memoryless determinacy over finite arenas, which is a by-product of their characterization through monotony and selectivity. It reduces the problem of memoryless determinacy over finite arenas to memoryless determinacy over finite *one-player* arenas.

> **Theorem 3.1.1** (One-to-two-player memoryless lift, finite arenas [GZ05, Corollary 7]) *Let $W \subseteq C^\omega$ be an objective. If memoryless strategies suffice for both $\mathcal{P}_1$ and $\mathcal{P}_2$ in their respective finite one-player arenas, then W is memoryless-determined over finite arenas.*

[GZ05]: Gimbert et al. (2005), *Games Where You Can Play Optimally Without Any Memory*

Such a "one-to-two-player lift" provides a neat and often easy way to prove that an objective admits memoryless optimal strategies without proving monotony and selectivity: proving it in the finite one-player arenas of the two players, which is generally easier as it boils down to graph reasoning, and then lifting the result to the general finite two-player arenas through the lift. We give an illustration of this result on the mean-payoff objective.

> **Example 3.1.2** Let $C = \mathbb{Q}$. We consider the mean-payoff objective $\mathsf{MP}^{\geq 0}$ (defined in Definition 2.4.12). This objective is memoryless-determined over finite arenas, which was first proved in [EM79]. We argue that it is easy to recover this result using Theorem 3.1.1.
>
> We take the point of view of $\mathcal{P}_1$ (due to the nature of objective $\mathsf{MP}^{\geq 0}$, $\mathcal{P}_2$ can be dealt with symmetrically). Let $\mathcal{A}$ be a *finite one-player* arena of $\mathcal{P}_1$. We show that $\mathcal{P}_1$ has a memoryless optimal strategy in $\mathcal{A}$. We consider the cycles of $\mathcal{A}$, and we say that a cycle is *non-negative* if the average of its colors is non-negative. Clearly, $\mathcal{P}_1$ can win for $\mathsf{MP}^{\geq 0}$ if a non-negative cycle is reachable from the initial vertex: $\mathcal{P}_1$ can simply reach the cycle and loop around it. To win with a memoryless strategy, we simply observe that a non-negative cycle always contains a non-negative *simple* cycle (i.e., not going twice through the same vertex). A *simple* cycle can be reached and looped around with a *memoryless* strategy. This describes a way to win with a memoryless strategy if there is a non-negative cycle. This reasoning is actually *complete*: if there is no non-negative cycle, then $\mathcal{P}_1$ simply cannot win for $\mathsf{MP}^{\geq 0}$ (its mean payoff will be upper bounded by the simple cycle with the highest average, but this average is negative). We have shown that given a fixed initial vertex of $\mathcal{A}$, if $\mathcal{P}_1$ can win, then $\mathcal{P}_1$ can win with a memoryless strategy. Formally, we must still argue that we can build a single memoryless strategy winning "uniformly" from all the vertices from which $\mathcal{P}_1$ has a winning strategy,

[EM79]: Ehrenfeucht et al. (1979), *Positional Strategies for Mean Payoff Games*

Note that the "simple cycle with the highest average" may not exist in an infinite arena.

which we do not do here but is reasonably straightforward (in particular, it will follow from results in Section 4.7).

These arguments show that memoryless strategies suffice for $\mathcal{P}_1$ in its finite one-player arenas, and the same reasoning works for $\mathcal{P}_2$ (using negative cycles instead). By Theorem 3.1.1, objective $\mathsf{MP}^{\geq 0}$ is memoryless-determined over finite arenas.

A natural question arises: which preference relations admit *finite-memory* optimal strategies? Surprisingly, whether an equivalent to Gimbert and Zielonka's characterization could be obtained in the finite-memory case or not has remained an open question up until recently. It is worth noticing that such an equivalent could be of great help in practice, especially if a *one-to-two-player lift* also holds: see for example [BMR+18; BHM+17; BHRR19], where proving that finite-memory strategies suffice in one-player games was fairly easy, in contrast to the high complexity of the two-player case — a lifting corollary could grant the two-player case for free!

We initially hoped that the following conjecture could be established: "if *finite-memory* strategies suffice for both $\mathcal{P}_1$ and $\mathcal{P}_2$ in their respective finite one-player arenas, they also suffice in all finite two-player arenas". Unfortunately, we found counterexamples to this conjecture in its more general form.

3.1.2 Lifting attempts for finite-memory determinacy

The goal of this section is to present a counterexample to the above conjecture; finite-memory determinacy over finite one-player games does not imply finite-memory determinacy over finite two-player games. This justifies the restriction of the notion of finite-memory determinacy to *arena-independent* finite-memory determinacy, for which we will prove a one-to-two-player lift in Chapter 4.

This counterexample is a variant of an energy objective. Let us assume that $C = \mathbb{Z}$, and that the objective of $\mathcal{P}_1$ is to create a play such that (*i*) the running sum of colors grows up to infinity (e.g., consider its lim inf to define it properly), *or* (*ii*) this running sum of colors takes value zero infinitely often. Formally, we consider the two objectives

$$W_1 = \{c_1 c_2 \ldots \in C^\omega \mid \liminf_{n \to \infty} \sum_{i=1}^n c_i = +\infty\},$$

$$W_2 = \{c_1 c_2 \ldots \in C^\omega \mid \sum_{i=1}^n c_i = 0 \text{ for infinitely many values of } n\},$$

and we define $W = W_1 \cup W_2$ as the objective of $\mathcal{P}_1$.

Proposition 3.1.3 *For objective W, both players have finite-memory optimal strategies in their respective finite one-player arenas, but $\mathcal{P}_1$ needs infinite*

memory to play optimally in some finite two-player arena.

Proof. Throughout the proof, we say that a cycle in an arena is a *zero* cycle if the sum of its colors is zero, a *positive* cycle if this sum is positive, and a *negative* cycle if this sum is negative.

We first consider a finite one-player arena of $\mathcal{P}_1$. In this arena, $\mathcal{P}_1$ can create a play ρ such that $\mathsf{col}^\omega(\rho) \in W_1$ if and only if there is a reachable positive cycle. In this case, $\mathcal{P}_1$ can win with a finite-memory strategy (simply reaching the cycle and then looping around it). If that is not possible, in order to win, $\mathcal{P}_1$ has to induce a play ρ such that $\mathsf{col}^\omega(\rho) = c_1 c_2 \ldots \in W_2$. As there are infinitely many indices for which the energy level is 0 but finitely many vertices in the arena, we can find two indices $k, l \geq 1$ such that $k < l$, $\mathsf{out}(e_k) = \mathsf{out}(e_l)$, $\sum_{i=1}^{k} c_i = 0$, and $\sum_{i=1}^{l} c_i = 0$. Notice in particular that $\sum_{i=k+1}^{l} c_i = 0$. Now, consider the play

$$\rho' = e_1 \ldots e_k (e_{k+1} \ldots e_l)^\omega,$$

with the sequence of edges $e_{k+1} \ldots e_l$ repeating ad infinitum (ρ' is a "lasso"). This is a valid play since $\mathsf{in}(e_{k+1}) = \mathsf{out}(e_k) = \mathsf{out}(e_l)$. Moreover, we have that $\mathsf{col}^\omega(\rho') \in W_2$ as after repeating m times the sequence $e_{k+1} \ldots e_l$, the sum of the colors equals $\sum_{i=1}^{k} c_i + m \cdot \sum_{i=k+1}^{l} c_i = 0 + m \cdot 0 = 0$. The play ρ' can be implemented with finite memory, as it consists of a finite prefix and a repeated finite sequence.

We now turn our attention to a finite one-player arena of $\mathcal{P}_2$; $\mathcal{P}_2$ wins by making a play ρ such that $\mathsf{col}^\omega(\rho) = c_1 c_2 \ldots \in C^\omega$ with

$$\liminf_{n \to \infty} \sum_{i=1}^{n} c_i < +\infty \wedge \sum_{i=1}^{n} c_i = 0 \text{ for at most finitely many values of } n.$$

If there is a reachable negative cycle in the arena, $\mathcal{P}_2$ can ensure to win by looping around it forever, and can therefore win with a finite-memory strategy. We now assume that there is no reachable negative cycle. As we did for $\mathcal{P}_1$, we show that if $\mathcal{P}_2$ can win the game in such an arena, then $\mathcal{P}_2$ can do so using finite memory.

If $\mathcal{P}_2$ can win, let $\rho = e_1 e_2 \ldots$ be a winning play for $\mathcal{P}_2$, i.e., $\mathsf{col}^\omega(\rho) = c_1 c_2 \ldots \in \overline{W_1} \cap \overline{W_2}$. Let v be a vertex visited infinitely often when ρ is played, and $m \geq 0$ be the first index such that $\mathsf{out}(e_m) = v$. We can decompose ρ into a finite prefix $e_1 \ldots e_m$ followed by an infinite sequence of cycles on v. Since there is no negative cycle, we cannot have that infinitely many of these cycles are positive, as this would imply that $\mathsf{col}^\omega(\rho) \in W_1$. Thus, infinitely many zero cycles from v are taken. As $\mathsf{col}^\omega(\rho) \in \overline{W_2}$, clearly, one of these cycles $e_{k+1} \ldots e_l$ (with $\mathsf{in}(e_{k+1}) = \mathsf{out}(e_l) = v$ and $\sum_{i=k+1}^{l} c_i = 0$) satisfies that for all n, $k + 1 \leq n \leq l$, it holds that $\sum_{i=1}^{n} c_i \neq 0$ (infinitely many cycles satisfy this condition, but we just pick one). This also implies that $\sum_{i=1}^{k} c_i \neq 0$, i.e., the history up to this cycle has a non-zero energy
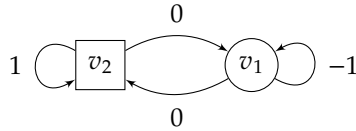
Figure 3.1: $\mathcal{P}_1$ (circle) needs infinite memory to win in this arena for objective W (by always resetting the sum of colors to zero by looping long enough on v_1 before going back to v_2).

level. Now, let us consider the play

$$\rho' = e_1 \ldots e_k (e_{k+1} \ldots e_l)^\omega,$$

with the sequence of edges $e_{k+1} \ldots e_l$ repeating ad infinitum (ρ' is a "lasso"). This is a valid play as $\text{in}(e_{k+1}) = \text{out}(e_l)$. Every time the cycle starts again, the running sum of colors is equal to the same value: $\sum_{i=1}^{k} c_i \neq 0$. Therefore, as the running sum of colors does not reach zero the first time the cycle is taken, and it also never reaches zero along the cycle, it can never reach zero after index $k - 1$. Hence, $\text{col}^\omega(\rho')$ is not in W, and ρ' is winning for $\mathcal{P}_2$. This play ρ' only needs finite memory to be implemented.

Now, consider the simple two-player game depicted in Figure 3.1. First, observe that $\mathcal{P}_1$ (circle) has an *infinite-memory* strategy to win: $\mathcal{P}_1$ should keep track of the running sum of colors (which is unbounded, hence the need for infinite memory). Whenever the current vertex is v_1, $\mathcal{P}_1$ should loop on v_1 up to the point where this sum hits zero, and then go to v_2. This strategy ensures victory because either $\mathcal{P}_2$ always goes back to v_1, in which case W_2 is satisfied, or $\mathcal{P}_2$ eventually loops forever on v_2, in which case W_1 is satisfied. It remains to argue that $\mathcal{P}_1$ has no *finite-memory* winning strategy in this game: whatever the finite amount of memory used by $\mathcal{P}_1$, $\mathcal{P}_2$ may loop on v_2 long enough as to exceed the bound up to which $\mathcal{P}_1$ can track the sum accurately; thus preventing $\mathcal{P}_1$ from resetting the sum to zero in v_1 infinitely often. □

This example proves that Gimbert and Zielonka's *one-to-two-player lift* cannot work in full generality in the finite-memory case. Informally, in the case of *memoryless* strategies, as in [GZ05], $\mathcal{P}_1$ is already doomed in one-player arenas in the absence of *monotony*: two prefixes to distinguish in order to play optimally can be "hardcoded" as different paths leading to the same vertex in a game arena, as if they were chosen by $\mathcal{P}_2$ in a two-player game. In the case of *finite-memory* strategies, however, the situation is different. In finite one-player arenas, the number of such paths that can be hardcoded in an arena is always bounded; hence finite memory might suffice to react, i.e., to keep track of which prefix is the current one and how to behave accordingly. However, in two-player arenas, $\mathcal{P}_2$ might create an *infinite* number of prefixes to distinguish (using a cycle), thus requiring $\mathcal{P}_1$ to use infinite memory to be able to do so. This is what happens in the example above: in any finite one-player game, the largest sum that $\mathcal{P}_1$ has to track is bounded, whereas $\mathcal{P}_2$ can make this sum arbitrarily large in some two-player arenas.

[GZ05]: Gimbert et al. (2005), *Games Where You Can Play Optimally Without Any Memory*

"Two prefixes to distinguish to play optimally" can be explained as two prefixes incomparable for the prefix preorder $\leq$ of W: they each have winning continuations that are not winning for the other prefix. In our example, the family of words $\{1^n \in C^* \mid n \geq 0\}$ are all incomparable for $\leq$.

3.2 Infinite game graphs

In Chapter 5, we will discuss finite-memory determinacy of games played on graphs that may have arbitrary cardinality. Before getting there, we discuss again the special case of *memoryless determinacy*, which is quite well-understood in the literature.

After showing an elegant characterization for memoryless determinacy of games played on *finite* arenas (Theorem 3.1.1), a legitimate question is: do we need in general more complex strategies to play optimally when arenas are allowed to be infinite? The answer turns out to be positive, which was already observed by Puterman in 1994 [Put94].

[Put94]: Puterman (1994), *Markov Decision Processes: Discrete Stochastic Dynamic Programming*

Example 3.2.1 We consider again objective $\mathsf{MP}^{\geq 0}$, but we now consider the *infinite* arena in Figure 3.2 (left) [Put94, Example 8.10.2]. Despite the fact that all colors are negative, $\mathcal{P}_1$ has a winning strategy from v_1: the idea is, for all $i \geq 1$, to loop on vertex v_i sufficiently many times to bring the mean payoff close to $-\frac{1}{i}$, and then move on to v_{i+1} and repeat. At the limit, the mean payoff is 0.

Concretely, it suffices to loop i times on vertex v_i before moving on to v_{i+1}. We show this by computing the mean color seen after n steps by aggregating what happens in each vertex, given that we make $i+1$ actions in vertex v_i (i times the self-loop with color $-\frac{1}{i}$ and 1 time the rightward edge with color -1). After n steps, if $n = (1+1)+(2+1)+\ldots+(k+1)+l$ with $0 \leq l \leq k+1$, the current mean payoff is

$$\frac{1}{n}\left(\sum_{i=1}^{k}(-\frac{1}{i}\cdot i - 1) - \frac{1}{k+1}\cdot l\right) = \frac{1}{n}\cdot\left(-2k - \frac{1}{k+1}\cdot l\right)$$

$$\leq \frac{1}{n}\cdot(-2k-1).$$

As n is quadratic in k, this expression indeed converges to 0 as n grows to infinity.

This winning strategy requires *infinite* memory to be implemented. No memoryless strategy is winning, as a memoryless strategy either stops forever in some vertex v_i and obtains a mean payoff of $-\frac{1}{i}$, or goes only to the right and obtains a mean payoff of -1. Hence, $\mathsf{MP}^{\geq 0}$ is not memoryless-determined over infinite arenas, even one-player ones.

We can generalize this reasoning to show that any finite-memory strategy with k states is losing. Such a strategy either stops forever in some vertex v_i and obtains a mean payoff of $-\frac{1}{i}$, or goes infinitely often to the right. In this second case, it can spend at most k steps in each vertex before moving to the right, hence color -1 is seen at least every k steps and the mean payoff then gets smaller than $-\frac{1}{k}$. Objective $\mathsf{MP}^{\geq 0}$ is not finite-memory-determined over infinite arenas either.

It might seem that we actively rely on the access to *infinitely many colors in the same arena* to build the example, which is obviously not possible
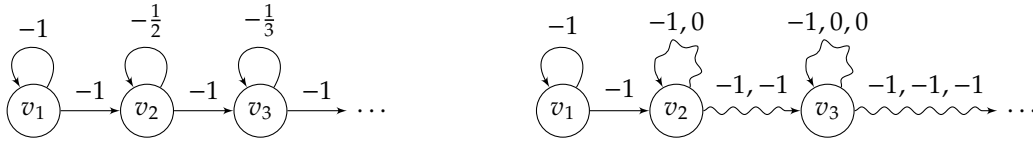
Figure 3.2: Infinite one-player arenas requiring infinite memory for objective $\mathsf{MP}^{\geq 0}$. The arena on the right uses only finitely many colors.

in finite arenas. We depict a slightly more involved arena in Figure 3.2 (right) that exhibits similar properties for the same reasons, but using only colors -1 and 0. In a vertex v_i, we simply replace color $-\frac{1}{i}$ by the sequence $-1, 0^{i-1}$ and the rightward -1 colors by the sequence $(-1)^i$.

When denoting words composed of numbers, we separate them with commas to avoid ambiguities (even though we do not do this for arbitrary words $c_1 c_2 \ldots$).

This example shows that there are fewer memoryless-determined objectives if we allow arenas of arbitrary cardinality. One might wonder: what objectives then remain memoryless-determined? We have seen that parity conditions are memoryless-determined over arenas of any cardinality (Theorem 2.6.3). We now discuss an elegant characterization by Colcombet and Niwiński [CN06] establishing that, under prefix-independence, there is actually nothing more than parity conditions.

[CN06]: Colcombet et al. (2006), *On the positional determinacy of edge-labeled games*

Theorem 3.2.2 ([CN06, Theorem 4]) *Let $W \subseteq C^\omega$ be a **prefix-independent** objective. If W is memoryless-determined (in games played on infinite arenas), then there is $n \in \mathbb{N}$ and a function $p \colon C \to \{0, \ldots, n\}$ such that*

$$w = c_1 c_2 \ldots \in W \iff \limsup_{i \to \infty} p(c_i) \text{ is even}.$$

In other words, up to renaming colors through a priority function p taking values in a finite set of natural numbers, W is a parity condition.

Using memoryless determinacy of parity conditions (Theorem 2.6.3), this result provides the missing implication of an equivalence between being a parity condition and being memoryless-determined (under prefix-independence assumption). As objective $\mathsf{MP}^{\geq 0}$ is prefix-independent but is not a parity condition, it is thus not surprising that it requires memory in some infinite arenas for at least one player, as shown in Example 3.2.1.

To prove formally that $\mathsf{MP}^{\geq 0}$ is not a parity condition, observe that $(-1, -1)^\omega$ is losing while $(-1, 1, 1)^\omega$ is winning, so the winning/losing status of words does not simply depend on the colors seen infinitely often.

Although not stated by Colcombet and Niwiński, we can also get a *one-to-two-player lift* for games played on infinite arenas from their proof technique. Indeed, their result can be obtained just by making an assumption about memory requirements in *one-player* arenas. This requires carefully observing that all the arenas they use in their proof are *one-player* arenas, except for [CN06, Lemma 9] which can be made to work with a one-player arena instead. This was already discussed by Kopczyński in his thesis [Kop08, Theorem 3.8].

[Kop08]: Kopczyński (2008), *Half-positional Determinacy of Infinite Games*

> **Corollary 3.2.3** (One-to-two-player memoryless lift, infinite arenas) *Let $W \subseteq C^\omega$ be a **prefix-independent** objective. If both players have memoryless optimal strategies in their respective one-player arenas (of any cardinality), then W is a parity condition (up to renaming the colors). In particular, W is memoryless-determined (over two-player arenas of any cardinality).*

Corollary 3.2.3 will be a special case of our results in Chapter 5, so we do not provide more proof details for it here.

This corollary makes it reasonable that the counterexample to the memoryless determinacy of $\mathsf{MP}^{\geq 0}$ in infinite arenas we exhibited in Example 3.2.1 was a one-player arena; it was guaranteed that there was a *one-player* arena in which some player did not have a memoryless optimal strategy.

Based on the knowledge of Theorem 3.2.2, we identified two questions that still needed answers and that fit our goals.

> ▶ What if the objective is not prefix-independent? There are many memoryless-determined objectives (even over infinite arenas) that are not parity conditions; for instance, $\mathsf{Reach}(a)$ is one of them. Can we find a characterization encompassing such objectives?
> ▶ What if we relax memoryless determinacy to (chromatic) finite-memory determinacy? Can we still obtain a nice characterization for this larger class of objectives?

We will give precise answers to both these questions in Chapter 5.

Characterization of arena-independent finite-memory determinacy

4

In Section 3.1, we discussed a characterization of memoryless determinacy by Gimbert and Zielonka in games played on finite graphs [GZ05], and what can be expected for a generalization to finite-memory determinacy. In this chapter, we establish a characterization of objectives that admit optimal strategies using *arena-independent* finite memory, providing an extension of the work of Gimbert and Zielonka to the finite-memory case. The adjective *arena-independent* refers to memory structures that can depend on the set of colors and the objective (with its possible parameters), but not on the precise game graph chosen to play the game. We prove an equivalent to their practical *one-to-two-player lift*: for a given objective, if both players have optimal arena-independent-finite-memory strategies in all finite one-player graphs, then it is also the case in all finite two-player graphs.

The contributions from this chapter are based on joint work with Patricia Bouyer and Stéphane Le Roux (both from Université Paris-Saclay, CNRS, ENS Paris-Saclay, LMF), Youssouf Oualhadj (Univ Paris Est Creteil, LACL), and Mickael Randour (F.R.S.-FNRS & Université de Mons) published in two papers: a conference version [BLO+20] in the proceedings of *CONCUR'20* and a more extensive journal version [BLO+22] in *Logical Methods in Computer Science*.

4.1 Introduction

Preference relations.   Up to now, we have discussed objectives encoded as sets of infinite words — such objectives are usually called *qualitative*, as a player either wins or loses. Inspired by games in economics, we can generalize this to *payoff functions* mapping infinite words to numerical values, and $\mathcal{P}_1$ can be seen as the "maximizer" player. The two formalisms are strongly linked: the classical decision problem for quantitative games is to fix a payoff threshold and ask if $\mathcal{P}_1$ has a strategy to guarantee it, essentially transforming the problem into a qualitative one (where the winning plays are all those achieving a payoff at least equal to the threshold). This is for instance what we did with the mean-payoff function, to which we have fixed threshold 0 in the examples (Examples 3.1.2 and 3.2.1). These quantitative objectives especially find meaning in the *reactive synthesis* motivation of considering zero-sum games: they allow describing complex, non-binary specifications where some quantity (e.g., energy consumption, response time) has to be maximized or minimized [BMR+18; BHM+17; BHRR19].

In this chapter, we borrow the formalism of Gimbert and Zielonka [GZ05] encompassing qualitative and quantitative objectives: we consider *preference relations* over infinite words. This general formalism permits encoding most classical game objectives, both qualitative and quantitative, and lets us reason in a well-founded framework under minimal assumptions.

Contributions.   We generalize Gimbert and Zielonka's results — characterization and one-to-two-player lift — to the case of *arena-independent* finite memory. That is, we encompass situations where the memory needed by the two players is *solely dependent on the preference relation* (e.g., colors, dimensions of weight vectors), and *not* on the game arena (i.e., number of vertices/edges). This restriction to arena-independent memory is natural given the counterexamples to a more general approach presented in Section 3.1.

Informally, our characterization can be stated as follows: given a preference relation and a memory structure $\mathcal{M}$, both players have finite-memory optimal strategies based on memory structure $\mathcal{M}$ in all finite game graphs if and only if the preference relation and its inverse are $\mathcal{M}$-monotone and $\mathcal{M}$-selective.

These last two concepts correspond intuitively to Gimbert and Zielonka's monotony and selectivity, *modulo a memory structure*. Recall that monotony and selectivity are related to the stability of the preference relation with regard to replacing prefixes and combining cycles, respectively. Our more general concepts of $\mathcal{M}$-*monotony* and $\mathcal{M}$-*selectivity* serve the same purpose, but they only compare sequences of colors deemed equivalent by the memory structure. For the sake of illustration, take selectivity: it implies that one has no interest in mixing different cycles of the game arena. For its

[BMR+18]: Bouyer et al. (2018), *Average-energy games*
[BHM+17]: Bouyer et al. (2017), *Bounding Average-Energy Games*
[BHRR19]: Bruyère et al. (2019), *Energy Mean-Payoff Games*
[GZ05]: Gimbert et al. (2005), *Games Where You Can Play Optimally Without Any Memory*

generalization, the memory structure is taken into account: $\mathcal{M}$-selectivity implies that one has no interest in mixing cycles of the game arena *that are read as cycles on the same state in the memory structure $\mathcal{M}$.*

Applicability. The arena-independent framework naturally includes memoryless-determined preference relations (the case studied in [GZ05]), for which the memory structure $\mathcal{M}_{\text{triv}}$ with a single state suffices for both players in all finite arenas. Beyond memoryless determinacy, a typical class of objectives that we cover is the one of ω-regular objectives: a deterministic parity automaton recognizing the objective suffices to play optimally for both players in all arenas (Theorem 2.7.11). However, a DPA (even a minimal one) is only an *upper* bound on the amount of memory needed for the derived ω-regular objective. This justifies the relevance of providing dedicated tools to understand the memory requirements of ω-regular objectives, which is not equivalent to providing tools to represent ω-regular objectives as automata.

[GZ05]: Gimbert et al. (2005), *Games Where You Can Play Optimally Without Any Memory*

Classes of ω-regular objectives whose memory requirements were studied previously contain, e.g., combinations of parity objectives [CHP07] (the memory depends on the number of objectives and the number of priorities), lower-bounded and upper-bounded energy objectives [BFL+08; BMR+18; BHM+17] (the memory depends on the bounds and the weights), Muller conditions [DJW97; Cas22] (the memory depends on the colors and on the sets in $\mathcal{F}$). On the contrary, combinations of lower-bounded energy objectives (with no upper bound) require *arena-dependent* memory [VCD+15; CRR14; JLS15], as shown in Example 2.6.13: the memory depends on the size of the arena in addition to the weights used in it. Such a setting falls outside the scope of our results.

[CHP07]: Chatterjee et al. (2007), *Generalized Parity Games*

[BFL+08]: Bouyer et al. (2008), *Infinite Runs in Weighted Timed Automata with Energy Constraints*

[BMR+18]: Bouyer et al. (2018), *Average-energy games*

[BHM+17]: Bouyer et al. (2017), *Bounding Average-Energy Games*

[DJW97]: Dziembowski et al. (1997), *How Much Memory is Needed to Win Infinite Games?*

[Cas22]: Casares (2022), *On the Minimisation of Transition-Based Rabin Automata and the Chromatic Memory Requirements of Muller Conditions*

Additional related works. We mention here works sharing the similar philosophy of trying to provide general conditions about objectives and games that guarantee "low" strategy complexity in games played on finite arenas.

[VCD+15]: Velner et al. (2015), *The complexity of multi-mean-payoff and multi-energy games*

[CRR14]: Chatterjee et al. (2014), *Strategy synthesis for multi-dimensional quantitative objectives*

[JLS15]: Jurdziński et al. (2015), *Fixed-Dimensional Energy Games are in Pseudo-Polynomial Time*

Following the same motivation as our work — the need to characterize (combinations of) objectives admitting finite-memory optimal strategies in finite arenas, Le Roux, Pauly, and Randour [LPR18] take another path: whereas our work permits lifting results from one-player games to two-player games, they provide a lifting from the single-objective case to the multi-objective one.

[LPR18]: Le Roux et al. (2018), *Extending Finite-Memory Determinacy by Boolean Combination of Winning Conditions*

Our work focuses on *deterministic turn-based* two-player games. Some sufficient conditions, orthogonal to our approach, were studied for *concurrent* games [LeR18; BLT22].

Many articles consider *stochastic* turn-based two-player games played on finite graphs (where transitions happen stochastically, and the goal is then usually to maximize the expected value of some payoff function). A

[LeR18]: Le Roux (2018), *Concurrent Games and Semi-Random Determinacy*

[BLT22]: Bouyer et al. (2022), *Finite-Memory Strategies in Two-Player Infinite Games*

line of work gives sufficient conditions for the existence of memoryless optimal strategies just for one player (i.e., for half-positionality): the same conditions were first proved sufficient in one-player stochastic games (often called *Markov decision processes*) [Gim07] and two-player deterministic games [Kop06], and then in two-player stochastic games [GK14] through more involved results [Mas15]. In [MSTW21] and in a new version of [GK14], under the same sufficient conditions, the authors provide a sufficient condition for finite-memory determinacy of the other player. We will compare this result to ours more extensively in Subsection 9.2.1, at the end of this thesis.

We have published an extension of the results from this chapter to stochastic games [BORV21a], building on previous unpublished work by Gimbert and Zielonka [GZ09]. To keep a consistent model throughout the thesis, we have chosen not to present this work in the thesis and to focus on the simpler deterministic version. We still discuss at a high level a description of the stochastic case at the end of the chapter, in Section 4.8.

Finally, in the same context of deterministic turn-based games, the one-to-two-player lift from this chapter was generalized later on by Kozachinskiy [Koz22b] to some constrained condition on the "arena-dependence" of the memory requirements in one-player games, by making finer observations along the same proof technique. We also come back to this result in Section 4.8.

Chapter structure. We extend qualitative objectives to the more general framework of preference relations and introduce some technical notions used in this chapter in Section 4.2. In Section 4.3, we introduce $\mathcal{M}$-monotony and $\mathcal{M}$-selectivity, generalizing concepts from [GZ05] to take into account memory structures. These are the core concepts of our characterization, i.e., the properties that preference relations must satisfy to admit optimal strategies based on $\mathcal{M}$. Section 4.4 is devoted to our results, and contains our characterization (Theorem 4.4.1) and the novel one-to-two-player lift (Theorem 4.4.4). We provide an application of our concepts in Subsection 4.4.2. The proof of the characterization is split into Section 4.5 (for the necessity of $\mathcal{M}$-monotony and $\mathcal{M}$-selectivity) and Section 4.6 (for the sufficient direction). The final two sections are orthogonal discussions. Section 4.7 is devoted to discussing the *uniformity* requirement of winning strategies; thanks to the results of this chapter, we can say more about the impact that requiring uniformity has on strategy complexity. We discuss some chosen related works in greater detail in Section 4.8.

[Gim07]: Gimbert (2007), *Pure Stationary Optimal Strategies in Markov Decision Processes*

[Kop06]: Kopczyński (2006), *Half-Positional Determinacy of Infinite Games*

[GK14]: Gimbert et al. (2014), *Submixing and Shift-Invariant Stochastic Games*

[Mas15]: Mashiah-Yaakovi (2015), *Correlated Equilibria in Stochastic Games with Borel Measurable Payoffs*

[MSTW21]: Mayr et al. (2021), *Simple Stochastic Games with Almost-Sure Energy-Parity Objectives are in NP and coNP*

[BORV21a]: Bouyer et al. (2021), *Arena-Independent Finite-Memory Determinacy in Stochastic Games*

[GZ09]: Gimbert et al. (2009), *Pure and Stationary Optimal Strategies in Perfect-Information Stochastic Games with Global Preferences*

[Koz22b]: Kozachinskiy (2022), *One-To-Two-Player Lifting for Mildly Growing Memory*

[GZ05]: Gimbert et al. (2005), *Games Where You Can Play Optimally Without Any Memory*

4.2 Additional preliminaries

We first extend notions defined about (qualitative) objectives to the more general framework of *preference relations*. We then introduce some links between non-deterministic automata and arenas, which will be technically helpful in the upcoming proofs.

4.2.1 Preference relations

We consider a generalization of qualitative objectives where a player may have a preference about arbitrarily many outcomes.

> **Definition 4.2.1** *A preference relation is a total preorder $\sqsubseteq \subseteq C^\omega \times C^\omega$ on C^ω.*

A *quantitative game* (extending qualitative games) is then a tuple $\mathcal{G} = (\mathcal{A}, \sqsubseteq)$ where $\mathcal{A}$ is an arena and $\sqsubseteq$ is a preference relation. In such a game, the objective of $\mathcal{P}_1$ is to create the best possible play with regard to $\sqsubseteq$ whereas the objective of $\mathcal{P}_2$ is to obtain the worst possible one. That is, $\mathcal{P}_2$ uses the inverse relation $\sqsubseteq^{-1}$ (such that $w \sqsubseteq^{-1} w'$ if and only if $w' \sqsubseteq w$). This corresponds to zero-sum games when using a quantitative framework.

Every qualitative game $\mathcal{G} = (\mathcal{A}, W)$ can be written as a quantitative game $(\mathcal{A}, \sqsubseteq_W)$, where $w \sqsubseteq_W w'$ if and only if $w \in \overline{W}$ or $w' \in W$.

> **Example 4.2.2** Consider the mean-payoff function MP, with $C = \mathbb{Q}$. We define a preference relation such that the goal of $\mathcal{P}_1$ is to create a word w maximizing the value of MP. Payoff function MP induces a natural preference relation $\sqsubseteq_{\mathsf{MP}}$ between sequences of colors as follows: for all $w, w' \in C^\omega, w \sqsubseteq_{\mathsf{MP}} w'$ if and only if $\mathsf{MP}(w) \leq \mathsf{MP}(w')$. Such quantitative games are zero-sum, hence $\mathcal{P}_2$ uses the natural inverse relation $\sqsubseteq_{\mathsf{MP}}^{-1}$; $\mathcal{P}_2$ is a minimizer player in the payoff formulation of these games.

Function MP was defined in Definition 2.4.12.

Given $w, w' \in C^\omega$, we write $w \sqsubset w'$ if we have $\neg(w' \sqsubseteq w)$ (we recall that the preorder is *total*). We extend the relation $\sqsubseteq$ to subsets of C^ω as follows: for $W, W' \subseteq C^\omega$,

$$W \sqsubseteq W' \iff \forall w \in W, \exists w' \in W', w \sqsubseteq w'.$$

We also write

$$W \sqsubset W' \iff \exists w' \in W', \forall w \in W, w \sqsubset w'.$$

Note that $W \sqsubset W'$ if and only if $\neg(W' \sqsubseteq W)$.

We sometimes compare words $w \in C^\omega$ with sets $W \subseteq C^\omega$ (writing $w \sqsubseteq W$), by simply identifying word w to its singleton language $\{w\}$.

> **Remark 4.2.3** For a preference relation $\sqsubseteq_W$ induced as explained above by a qualitative objective $W \subseteq C^\omega$, observe that for $W_1, W_2 \subseteq C^\omega$,
>
> $$W_1 \sqsubseteq_W W_2 \Longleftrightarrow (W_1 \cap W = \emptyset \wedge W_2 \neq \emptyset) \vee W_2 \cap W \neq \emptyset, \text{ and}$$
> $$W_1 \sqsubset_W W_2 \Longleftrightarrow W_1 \cap W = \emptyset \wedge W_2 \cap W \neq \emptyset.$$

In this new quantitative framework, we must define again what it means for a strategy to be *optimal*. Let $\mathcal{G} = (\mathcal{A}, \sqsubseteq)$ be a quantitative game on arena $\mathcal{A} = (V, V_1, V_2, E)$. Given a strategy σ_1 of $\mathcal{P}_1$ on $\mathcal{A}$ and a vertex $v \in V$, we define

$$\mathsf{UCol}_\sqsubseteq(\mathcal{A}, v, \sigma_1) = \{w \in C^\omega \mid \exists w' \in \mathsf{col}^\omega(\mathsf{Plays}(\mathcal{A}, v, \sigma_1)), w' \sqsubseteq w\}.$$

Intuitively, $\mathsf{UCol}_\sqsubseteq$ represents the *upward closure* of sequences of colors consistent with a strategy with respect to the preference relation $\sqsubseteq$.

Taking the standpoint of $\mathcal{P}_1$, we say that a strategy σ_1 of $\mathcal{P}_1$ on $\mathcal{A}$ is *at least as good as strategy σ_1' of $\mathcal{P}_1$ from vertex $v \in V$* if

$$\mathsf{UCol}_\sqsubseteq(\mathcal{A}, v, \sigma_1) \subseteq \mathsf{UCol}_\sqsubseteq(\mathcal{A}, v, \sigma_1').$$

Intuitively, σ_1 is at least as good as σ_1' if the "worst-case" plays consistent with σ_1 are at least as good as the ones consistent with σ_1'. The UCol operator is useful to define this notion properly even when there is no "worst-case" play for a strategy (i.e., if the infimum used in the classical quantitative setting is not reached). Similar notions have been used before, e.g., in [LeR13].

[LeR13]: Le Roux (2013), *Infinite sequential Nash equilibrium*

Symmetrically for $\mathcal{P}_2$, we define

$$\mathsf{DCol}_\sqsubseteq(\mathcal{A}, v, \sigma_2) = \{w \in C^\omega \mid \exists w' \in \mathsf{col}^\omega(\mathsf{Plays}(\mathcal{A}, v, \sigma_2)), w \sqsubseteq w'\},$$

and we say that a strategy σ_2 of $\mathcal{P}_2$ on $\mathcal{A}$ is *at least as good as strategy σ_2' from vertex $v \in V$* if

$$\mathsf{DCol}_\sqsubseteq(\mathcal{A}, v, \sigma_2) \subseteq \mathsf{DCol}_\sqsubseteq(\mathcal{A}, v, \sigma_2').$$

Now, we say that a strategy σ_ℓ of $\mathcal{P}_\ell$ on $\mathcal{A}$ is *as good as possible from $v \in V$* if it is at least as good as every strategy σ_ℓ' of $\mathcal{P}_\ell$ from v. We extend this notation to subsets of vertices: we say that a strategy σ_ℓ is *optimal from $V' \subseteq V$* if it is as good as possible from every $v \in V'$, and we simply say *optimal* to mean optimal from V.

4.2.2 Nash equilibria

We use *Nash equilibria* [Nas51; OR94] as tools to establish the existence of optimal strategies in some of our proofs. Let $\mathcal{G} = (\mathcal{A}, \sqsubseteq)$ be a quantitative game on arena $\mathcal{A} = (V, V_1, V_2, E)$. Formally, a *Nash equilibrium (NE) from a*

[Nas51]: Nash (1951), *Non-cooperative Games*
[OR94]: Osborne et al. (1994), *A course in game theory*

vertex $v \in V$ is a couple of strategies (σ_1, σ_2) of $\mathcal{P}_1$ and $\mathcal{P}_2$ such that, for all strategies σ_1' of $\mathcal{P}_1$, and all strategies σ_2' of $\mathcal{P}_2$,

$$\mathsf{col}^\omega(\mathsf{Plays}(\mathcal{A}, v, \sigma_1', \sigma_2)) \sqsubseteq \mathsf{col}^\omega(\mathsf{Plays}(\mathcal{A}, v, \sigma_1, \sigma_2))$$
$$\sqsubseteq \mathsf{col}^\omega(\mathsf{Plays}(\mathcal{A}, v, \sigma_1, \sigma_2')). \quad (4.1)$$

This means that neither $\mathcal{P}_1$ nor $\mathcal{P}_2$ has any interest in deviating from their respective strategy.

We take a moment to discuss the link between *optimal strategies* and *Nash equilibria* in our specific context of zero-sum games. Both notions seem related, and indeed, in [GZ05], Gimbert and Zielonka did choose Equation (4.1) — i.e., the definition of a Nash equilibrium — as their definition of *a pair of optimal strategies*. This suggests that both notions coincide. However, they do not in full generality, as we discuss in the following.

As stated before, our goal is to characterize preference relations that admit *finite-memory optimal strategies*, but Nash equilibria will serve as tools in our endeavor. Let us establish two interesting properties of Nash equilibria in zero-sum games.

First, it is possible to "mix" different Nash equilibria.

> **Lemma 4.2.4** *Let $\mathcal{G} = (\mathcal{A}, \sqsubseteq)$ be a quantitative game on arena $\mathcal{A} = (V, V_1, V_2, E)$, and let $v \in V$ be a vertex. Let (σ_1^a, σ_2^a) and (σ_1^b, σ_2^b) be two Nash equilibria from v. Then, (σ_1^a, σ_2^b) is also a Nash equilibrium from v.*

Proof. We need to prove that for all pairs of strategies σ_1', σ_2' of $\mathcal{P}_1$ and $\mathcal{P}_2$,

$$\mathsf{col}^\omega(\mathsf{Plays}(\mathcal{A}, v, \sigma_1', \sigma_2^b)) \sqsubseteq \mathsf{col}^\omega(\mathsf{Plays}(\mathcal{A}, v, \sigma_1^a, \sigma_2^b))$$
$$\sqsubseteq \mathsf{col}^\omega(\mathsf{Plays}(\mathcal{A}, v, \sigma_1^a, \sigma_2')). \quad (4.2)$$

Since (σ_1^a, σ_2^a) is an NE, we know that

$$\mathsf{col}^\omega(\mathsf{Plays}(\mathcal{A}, v, \sigma_1^b, \sigma_2^a)) \sqsubseteq \mathsf{col}^\omega(\mathsf{Plays}(\mathcal{A}, v, \sigma_1^a, \sigma_2^a))$$
$$\sqsubseteq \mathsf{col}^\omega(\mathsf{Plays}(\mathcal{A}, v, \sigma_1^a, \sigma_2^b)),$$

instantiating σ_1' and σ_2' to σ_1^b and σ_2^b respectively in Equation (4.1). Similarly, since (σ_1^b, σ_2^b) is an NE, we know that

$$\mathsf{col}^\omega(\mathsf{Plays}(\mathcal{A}, v, \sigma_1^a, \sigma_2^b)) \sqsubseteq \mathsf{col}^\omega(\mathsf{Plays}(\mathcal{A}, v, \sigma_1^b, \sigma_2^b))$$
$$\sqsubseteq \mathsf{col}^\omega(\mathsf{Plays}(\mathcal{A}, v, \sigma_1^b, \sigma_2^a)),$$

instantiating σ_1' and σ_2' to σ_1^a and σ_2^a respectively in Equation (4.1).

From the last two equations, we obtain that all four sequences of colors are equivalent for $\sqsubseteq$ as the inequalities form a cycle. Hence, since Equation (4.1)

holds for (σ_1^a, σ_2^a) and (σ_1^b, σ_2^b), and since $\mathsf{col}^\omega(\mathsf{Plays}(\mathcal{A}, v, \sigma_1^a, \sigma_2^a))$ is equivalent for $\sqsubseteq$ to $\mathsf{col}^\omega(\mathsf{Plays}(\mathcal{A}, v, \sigma_1^a, \sigma_2^a))$ and $\mathsf{col}^\omega(\mathsf{Plays}(\mathcal{A}, v, \sigma_1^b, \sigma_2^b))$, Equation (4.2) is satisfied. $\qquad\square$

> **Remark 4.2.5** Lemma 4.2.4 relies on the assumption that we consider
> *zero-sum* games, that is, $\mathcal{P}_2$ uses the inverse preference relation $\sqsubseteq^{-1}$. In
> our general proof scheme for this chapter, usage of Lemma 4.2.4 in
> Proposition 4.6.3 is the single argument preventing us from considering
> *non-zero-sum* games, where $\mathcal{P}_1$ and $\mathcal{P}_2$ use two different, unrelated,
> preference relations.

We now establish that Nash equilibria induce optimal strategies in our
zero-sum context.

> **Lemma 4.2.6** *Let $\mathcal{G} = (\mathcal{A}, W)$ be a quantitative game on arena $\mathcal{A} =
> (V, V_1, V_2, E)$, and let $v \in V$ be a vertex. Let (σ_1, σ_2) be a Nash equilibrium from v. Then, both σ_1 and σ_2 are as good as possible from v.*

Proof. We prove the statement for σ_1 (it works symmetrically for σ_2).
Consider the rightmost inequality of Equation (4.1). From it and the
elementary Lemma 2.3.3, we deduce that

$$\mathsf{UCol}_\sqsubseteq(\mathcal{A}, v, \sigma_1) = \{w \in C^\omega \mid \mathsf{col}^\omega(\mathsf{Plays}(\mathcal{A}, v, \sigma_1, \sigma_2)) \sqsubseteq w\}. \qquad (4.3)$$

Indeed, σ_2 is a *best response* [OR94] to σ_1.

We claim that σ_1 is at least as good as every strategy from v, hence that σ_1
is as good as possible from v. Let σ_1' be a strategy of $\mathcal{P}_1$ on $\mathcal{A}$. We need
to prove that $\mathsf{UCol}_\sqsubseteq(\mathcal{A}, v, \sigma_1) \subseteq \mathsf{UCol}_\sqsubseteq(\mathcal{A}, v, \sigma_1')$. Let $w \in \mathsf{UCol}_\sqsubseteq(\mathcal{A}, v, \sigma_1)$.
From Equation (4.3) and the leftmost inequality of Equation (4.1), we have
$\mathsf{col}^\omega(\mathsf{Plays}(\mathcal{A}, v, \sigma_1', \sigma_2)) \sqsubseteq w$. Hence, by definition, $w \in \mathsf{UCol}_\sqsubseteq(\mathcal{A}, v, \sigma_1')$.
$\qquad\square$

[OR94]: Osborne et al. (1994),
A course in game theory

> **Remark 4.2.7** As noted above, optimal strategies do not always coincide with Nash equilibria. They do coincide for *determined* preference
> relations. Up to now, we have only defined *determinacy* for qualitative objectives (Definition 2.4.4). We discuss briefly determinacy of preference
> relations.
>
> The traditional formulation of determinacy for real payoff functions
> $C^\omega \to \mathbb{R}$ requires the highest outcome that $\mathcal{P}_1$ can get close to (sup inf)
> to be equal to the lowest outcome that $\mathcal{P}_2$ can get close to (inf sup) — the
> game is then said to have a *value* [OR94]. Here, as we have not assumed
> much on preference relation $\sqsubseteq$ (which is a total preorder), we cannot
> assume the existence of infima and suprema (the concepts of $\mathsf{UCol}_\sqsubseteq$ and
> $\mathsf{DCol}_\sqsubseteq$ still imitate as much as possible the classical sup inf and inf sup
> formulations).
> Another less common (but equivalent [Ohl21, Lemma 3]) formulation

[Ohl21]: Ohlmann (2021),
*Monotonic graphs for parity and
mean-payoff games*

for determinacy of payoff functions is to require that all (qualitative) *threshold objectives* are determined. For a preference relation $\sqsubseteq$, the threshold objectives can be defined as the objectives

$$W_w = \{w' \in C^\omega \mid w \sqsubseteq w'\}$$

for each $w \in C^\omega$. For reasonable preference relations, by Borel determinacy [Mar75], all threshold objectives are indeed determined. Under this definition of determinacy, pairs of optimal strategies induce Nash equilibria. Hence, both notions coincide in most cases of interest. This discussion is still needed to understand the link between optimal strategies of the individual players and Nash equilibria (Gimbert and Zielonka only define the latter as *pairs of optimal strategies* [GZ05]). It is also useful to have a notion of optimality of a strategy that does not involve instantiating a strategy of the opponent.

[Mar75]: Martin (1975), *Borel determinacy*

[GZ05]: Gimbert et al. (2005), *Games Where You Can Play Optimally Without Any Memory*

Remark 4.2.8 In quantitative games played on one-player arenas, the two visions — optimal strategies and Nash equilibria — coincide.

4.2.3 Product arenas

Our main proof technique in this chapter is to perform a proof by induction on finite arenas to prove the existence of optimal strategies based on some memory structure. It turns out that a nice way to do this is to start by augmenting arenas with information from the memory structure, and only then start the induction on the class of arenas with enough information. This "augmenting" step can be formulated as a *product between an arena and a memory structure*.

Let $\mathcal{A} = (V, V_1, V_2, E)$ be an arena and $\mathcal{M} = (M, m_{\text{init}}, \alpha_{\text{upd}})$ be a memory structure. We define their *product* $\mathcal{A} \ltimes \mathcal{M}$ as the arena (V', V_1', V_2', E') where $V' = V \times M$, $V_1' = V_1 \times M$, $V_2' = V_2 \times M$, and $E' \subseteq V' \times C \times V'$ is such that $((v_1, m_1), c, (v_2, m_2)) \in E'$ if and only if $(v_1, c, v_2) \in E$ and $\alpha_{\text{upd}}(m_1, c) = m_2$. That is, the memory is updated according to the colors of the edges in E.

We use here an asymmetric symbol $\ltimes$ to emphasize that $\mathcal{A}$ and $\mathcal{M}$ are distinct objects, as opposed to the product of memory structures using symbol $\otimes$ (Definition 2.5.4).

A strategy on $\mathcal{A}$ based on memory $\mathcal{M}$ can be equivalently seen as a memoryless strategy on the product arena $\mathcal{A} \ltimes \mathcal{M}$. Albeit quite intuitive and part of folklore, we make this correspondence formal in the following result.

Lemma 4.2.9 *Let $\mathcal{G} = (\mathcal{A}, \sqsubseteq)$ be a quantitative game on arena $\mathcal{A} = (V, V_1, V_2, E)$. Let $\mathcal{M} = (M, m_{\text{init}}, \alpha_{\text{upd}})$ be a memory structure. For σ_ℓ a strategy of $\mathcal{P}_\ell$ on $\mathcal{A}$ based on $\mathcal{M}$ and $\alpha_{\text{nxt}} \colon V_\ell \times M \to E$ its next-action function, we have that σ_ℓ is optimal in $\mathcal{G}$ if and only if α_{nxt} (seen as a memoryless strategy) is optimal in $\mathcal{G}' = (\mathcal{A} \ltimes \mathcal{M}, \sqsubseteq)$ from $V \times \{m_{\text{init}}\}$.*

Proof. We first aim to define a bijection $\mathcal{H}\colon \mathsf{Hists}(\mathcal{A}) \to \mathsf{Hists}(\mathcal{A} \ltimes \mathcal{M}, V \times \{m_{\mathsf{init}}\})$. Let $\gamma = e_1 \ldots e_n \in \mathsf{Hists}(\mathcal{A})$, with $e_i = (v_{i-1}, c_i, v_i)$. We set $m_0 = m_{\mathsf{init}}$, and for $1 \leq i \leq n$, we set $m_i = \alpha_{\mathsf{upd}}(m_{i-1}, c_i)$. We define $e_i' = ((v_{i-1}, m_{i-1}), c_i, (v_i, m_i))$, and $\mathcal{H}(\gamma) = e_1' \ldots e_n'$. Notice that $\mathsf{col}^*(\mathcal{H}(\gamma)) = \mathsf{col}^*(\gamma)$. Furthermore, $\mathcal{H}$ is bijective; as the initial state of the memory m_{init} is fixed and the memory structure is deterministic, the memory states added to γ to obtain $\mathcal{H}(\gamma)$ are uniquely determined.

We now show that there is a correspondence between strategies of $\mathcal{P}_\ell$ on $\mathcal{A}$ and on $\mathcal{A} \ltimes \mathcal{M}$ through a bijection f: intuitively, augmenting the arena with memory structure $\mathcal{M}$ allows some strategies to be played using less memory, but does not fundamentally change each player's possibilities. For σ_ℓ a strategy of $\mathcal{P}_\ell$ on $\mathcal{A}$ and $\gamma' \in \mathsf{Hists}_\ell(\mathcal{A} \ltimes \mathcal{M})$ with $\mathsf{in}(\gamma') \in V \times \{m_{\mathsf{init}}\}$ and $\mathsf{out}(\gamma') = (v, m) \in V_\ell \times M$, if $\sigma_\ell(\mathcal{H}^{-1}(\gamma')) = (v, c, v')$, we define $f(\sigma_\ell)(\gamma') = ((v, m), c, (v', \alpha_{\mathsf{upd}}(m, c)))$. The histories induced by strategies σ_ℓ and $f(\sigma_\ell)$ correspond: if $\gamma = \mathcal{H}^{-1}(\gamma')$, then we have

$$\mathcal{H}(\gamma \cdot \sigma_\ell(\gamma)) = \gamma' \cdot (f(\sigma_\ell)(\gamma')). \tag{4.4}$$

We are only interested in the behavior of strategies on $\mathcal{A} \ltimes \mathcal{M}$ from histories γ' with $\mathsf{in}(\gamma') \in V \times \{m_{\mathsf{init}}\}$ (in what follows, we only consider histories and plays of $\mathcal{A} \ltimes \mathcal{M}$ starting in such vertices). If we restrict the image of f to the set of functions $\sigma_\ell'\colon \mathsf{Hists}_\ell(\mathcal{A} \ltimes \mathcal{M}, V \times \{m_{\mathsf{init}}\}) \to E'$, then f is a bijection.

We now show a second fact related to f: we have that for all $v \in V$, for all pairs σ_1, σ_2 of strategies of $\mathcal{P}_1$ and $\mathcal{P}_2$ on $\mathcal{A}$,

$$\mathsf{col}^\omega\big(\mathsf{Plays}(\mathcal{A}, v, \sigma_1, \sigma_2)\big) = \mathsf{col}^\omega\big(\mathsf{Plays}(\mathcal{A} \ltimes \mathcal{M}, (v, m_{\mathsf{init}}), f(\sigma_1), f(\sigma_2))\big).$$

This can easily be proved by induction using Equation (4.4). Indeed, at each step, both strategy σ_1 (resp. σ_2) and strategy $f(\sigma_1)$ (resp. $f(\sigma_2)$) pick an edge with the same color. Using Lemma 2.3.3 about consistent plays and bijectivity of f, we have that

$$\mathsf{col}^\omega\big(\mathsf{Plays}(\mathcal{A}, v, \sigma_1)\big) = \mathsf{col}^\omega\big(\mathsf{Plays}(\mathcal{A} \ltimes \mathcal{M}, (v, m_{\mathsf{init}}), f(\sigma_1))\big). \tag{4.5}$$

We finish the proof taking the point of view of $\mathcal{P}_1$; the proof is symmetric for $\mathcal{P}_2$. Let σ_1 be a strategy of $\mathcal{P}_1$ on $\mathcal{A}$. The definition of $\mathsf{UCol}_\sqsubseteq$ and Equation (4.5) imply that for all vertices $v \in V$,

$$
\begin{aligned}
&\mathsf{UCol}_\sqsubseteq(\mathcal{A}, v, \sigma_1) \\
&= \{w \in C^\omega \mid \exists w' \in \mathsf{col}^\omega\big(\mathsf{Plays}(\mathcal{A}, v, \sigma_1)\big), w' \sqsubseteq w\} \\
&= \{w \in C^\omega \mid \exists w' \in \mathsf{col}^\omega\big(\mathsf{Plays}(\mathcal{A} \ltimes \mathcal{M}, (v, m_{\mathsf{init}}), f(\sigma_1))\big), w' \sqsubseteq w\} \\
&= \mathsf{UCol}_\sqsubseteq(\mathcal{A} \ltimes \mathcal{M}, (v, m_{\mathsf{init}}), f(\sigma_1)). \qquad\qquad (4.6)
\end{aligned}
$$

Using Equation (4.6), we obtain that a strategy σ_1 is optimal in $\mathcal{G} = (\mathcal{A}, \sqsubseteq)$ if and only if $f(\sigma_1)$ is optimal in $\mathcal{G}' = (\mathcal{A} \ltimes \mathcal{M}, \sqsubseteq)$ from $V \times \{m_{\mathsf{init}}\}$.

We now assume that σ_1 is based on $\mathcal{M}$ and that α_{nxt} is its next-action function. Formally, we want to transform α_{nxt} into a proper memoryless strategy on $\mathcal{A} \ltimes \mathcal{M}$. This can be done through the bijection f, yielding the memoryless strategy $\alpha'_{\text{nxt}} = f(\sigma_1)$, which corresponds exactly to α_{nxt} interpreted over the product arena, and well-defined for all histories starting in $V \times \{m_{\text{init}}\}$. In particular, σ_1 is optimal in $\mathcal{G}$ if and only if $f(\sigma_1) = \alpha'_{\text{nxt}}$ is optimal in $\mathcal{G}'$ from $V \times \{m_{\text{init}}\}$. $\qquad\square$

> **Remark 4.2.10** Lemma 4.2.9 can be restated in terms of Nash equilibria, using a similar reasoning.

4.2.4 Arena induced by a non-deterministic finite automaton

We relate the words that can be induced in an arena with the language recognized by an automaton. We will need *non-deterministic* automata here, in order to emulate the fact that in an arena, there can be multiple outgoing edges with the same color from the same vertex.

A *non-deterministic finite automaton (NFA)* is a tuple $\mathcal{N} = (Q, C, \Delta, Q_{\text{init}}, F)$, where Q is a finite set of *states*, C is the set of colors, $\Delta \subseteq Q \times C \times Q$ is a finite set of *transitions*, $Q_{\text{init}} \subseteq Q$ is a set of *initial states*, and $F \subseteq Q$ is a set of *final states*. Given a state $q \in Q$ and a finite word $w \in C^*$, we denote by $\Delta^*(q, w)$ the set of states that can be reached from q after reading w. Without loss of generality, we assume all NFAs to be *coaccessible*, i.e., for all $q \in Q$, there exists $w \in C^*$ such that $\Delta^*(q, w) \cap F \neq \emptyset$.

Let $K \subseteq C^*$ be a language of *finite* words. We denote by $\mathsf{Prefs}(K)$ the set of all *prefixes* of the words in K. We define a set of infinite words derived from K:

$$[K] = \{w = c_1 c_2 \ldots \in C^\omega \mid \forall n \geq 1, c_1 \ldots c_n \in \mathsf{Prefs}(K)\},$$

which contains all infinite words for which every finite prefix is a prefix of a word in K. Intuitively, if K is regular, $[K]$ is the language of infinite words that correspond to infinite paths that can always branch and reach a final state on an NFA for K: we will formalize this in Lemma 4.2.12. Given a finite word $w \in C^*$ and a language $K \subseteq C^*$, we write wK for their concatenation, i.e., the language $wK = \{w' = ww'' \mid w'' \in K\} \subseteq C^*$.

The following observation, already noted in [GZ05], will prove useful.

> **Lemma 4.2.11** *Let $K_1, K_2 \subseteq C^*$. Then $[K_1 \cup K_2] = [K_1] \cup [K_2]$.*

Proof. Let $w \in [K_1 \cup K_2]$. Every finite prefix of w is in $\mathsf{Prefs}(K_1 \cup K_2)$. Assume w.l.o.g. that infinitely many prefixes of w are in $\mathsf{Prefs}(K_1)$. This implies that *all* prefixes of w are in $\mathsf{Prefs}(K_1)$ (intuitively, because there is a continuity in the prefix relation). Hence, $w \in [K_1] \cup [K_2]$.

Set $[K]$ is very close to the more classical *limit language* $\overrightarrow{K}$ [PP04]. The difference is that we require here that all prefixes of the infinite words are *prefixes of words of K*, while the limit language has the stronger requirement that infinitely many prefixes of the infinite word are *words of K*. For example, if $K = a^* b$, $[K] = \{a^\omega\}$, but $\overrightarrow{K} = \emptyset$.

[PP04]: Perrin et al. (2004), *Infinite words – automata, semigroups, logic and games*

[GZ05]: Gimbert et al. (2005), *Games Where You Can Play Optimally Without Any Memory*

Now, let $w \in [K_1] \cup [K_2]$. If $w \in [K_1]$ (resp. $[K_2]$), every finite prefix of w is in $\mathsf{Prefs}(K_1)$ (resp. $\mathsf{Prefs}(K_2)$), so in particular it is in $\mathsf{Prefs}(K_1 \cup K_2)$. Hence, $w \in [K_1 \cup K_2]$. □

Let $\mathcal{N} = (Q, C, \Delta, Q_{\mathsf{init}}, F)$ be an NFA. We say that a state $q \in Q$ is *essential* if there exists an infinite path in $\mathcal{N}$ starting in q. Let $Q_{\mathsf{ess}} = \{q \in Q \mid q \text{ is essential}\}$. We define the corresponding *finite one-player arena* $\mathsf{Arena}(\mathcal{N}) = (V = Q_{\mathsf{ess}}, V_1 = V, V_2 = \emptyset, E \subseteq Q_{\mathsf{ess}} \times C \times Q_{\mathsf{ess}})$, where $e = (q, c, q') \in E$ if $(q, c, q') \in \Delta$. Intuitively, $\mathsf{Arena}(\mathcal{N})$ transforms $\mathcal{N}$ into a *non-blocking* arena thanks to the restriction to essential states.

We may now state formally the link between $[K]$ and an underlying NFA for K. Our proof is similar to [GZ05, Lemma 4].

[GZ05]: Gimbert et al. (2005), *Games Where You Can Play Optimally Without Any Memory*

> **Lemma 4.2.12** *Let $\mathcal{N} = (Q, C, \Delta, Q_{\mathsf{init}}, F)$ be a (coaccessible) NFA recognizing the regular language $K \subseteq C^*$. Let $Q'_{\mathsf{init}} = Q_{\mathsf{init}} \cap Q_{\mathsf{ess}}$. The following equality holds:*
>
> $$[K] = \mathsf{col}^\omega\big(\mathsf{Plays}(\mathsf{Arena}(\mathcal{N}), Q'_{\mathsf{init}})\big).$$
>
> *In particular, $[K]$ is non-empty if and only if there exists an essential initial state in $\mathcal{N}$.*

Proof. If Q'_{init} is empty, the equality holds: $\mathsf{col}^\omega\big(\mathsf{Plays}(\mathsf{Arena}(\mathcal{N}), Q'_{\mathsf{init}})\big)$ and $[K]$ are both empty. Hence, from now on, we assume $Q'_{\mathsf{init}} \neq \emptyset$.

We start with the left-to-right inclusion. Let $w = c_1 c_2 \ldots \in [K]$. We first prove that for all $n \geq 1$, it holds that

$$c_1 \ldots c_n \in \mathsf{col}^*\big(\mathsf{Hists}(\mathsf{Arena}(\mathcal{N}), Q'_{\mathsf{init}})\big).$$

We assume on the contrary that there exists $n \geq 1$ such that

$$c_1 \ldots c_n \notin \mathsf{col}^*\big(\mathsf{Hists}(\mathsf{Arena}(\mathcal{N}), Q'_{\mathsf{init}})\big).$$

As $\mathsf{Arena}(\mathcal{N})$ is a restriction of the states of $\mathcal{N}$ to Q_{ess}, this means that no matter how $c_1 \ldots c_n$ is read on $\mathcal{N}$, it goes through a state in $Q \setminus Q_{\mathsf{ess}}$. As there is no infinite path from these states, this contradicts that $w \in [K]$; there cannot be arbitrarily long prefixes starting with $c_1 \ldots c_n$.

We now use the property that we have just proved along with Kőnig's lemma to show that $w \in \mathsf{col}^\omega\big(\mathsf{Plays}(\mathsf{Arena}(\mathcal{N}), Q'_{\mathsf{init}})\big)$. We build a forest $\mathcal{F}$ of trees. The nodes of $\mathcal{F}$ are paths $\gamma \in \mathsf{Hists}(\mathsf{Arena}(\mathcal{N}), Q'_{\mathsf{init}})$ such that $\mathsf{col}^*(\gamma)$ is a prefix of w and $\mathsf{in}(\gamma) \in Q'_{\mathsf{init}}$. For every $q \in Q'_{\mathsf{init}}$, there is one tree in $\mathcal{F}$ whose root is the empty path λ_q. There is a transition from a node γ to a node γ' if there exists $e \in E$ such that $\gamma \cdot e = \gamma'$. As there is at least one node for each prefix $c_1 \ldots c_n$, (at least) one of the trees of $\mathcal{F}$ must be infinite. Moreover, $\mathcal{F}$ is finitely branching, since $\mathcal{N}$ has finitely many transitions. By Kőnig's lemma, we obtain that there must be an infinite path ρ starting from a root λ_q for some $q \in Q'_{\mathsf{init}}$. By construction, $\mathsf{col}^\omega(\rho) = w$, so $w \in \mathsf{col}^\omega\big(\mathsf{Plays}(\mathsf{Arena}(\mathcal{N}), Q'_{\mathsf{init}})\big)$.

Kőnig's lemma [Kőn27] (instantiated for trees) states that in any finitely branching tree with infinitely many nodes, there is an infinite path.

[Kőn27]: Kőnig (1927), *Über eine Schlussweise aus dem Endlichen ins Unendliche*

We now prove the right-to-left inclusion. Let

$$\rho = e_1 e_2 \ldots \in \mathsf{Plays}(\mathsf{Arena}(\mathcal{N}), Q'_{\mathsf{init}}).$$

For $n \geq 1$, the word $\mathsf{col}^*(e_1 \ldots e_n)$ is the color of a path in $\mathcal{N}$, since every edge of $\mathsf{Arena}(\mathcal{N})$ corresponds to a transition of $\mathcal{N}$. As $\mathcal{N}$ is coaccessible, there is a path in $\mathcal{N}$ from the state corresponding to $\mathsf{out}(e_n)$ to a final state in F. Thus, the word $\mathsf{col}^*(e_1 \ldots e_n)$ is a prefix of an accepted word of $\mathcal{N}$, i.e., a prefix of a word in K; as this holds for all $n \geq 1$, we obtain that $\mathsf{col}^\omega(\rho) \in [K]$. □

We are now able to establish our characterization of preference relations admitting arena-independent finite-memory optimal strategies, i.e., optimal strategies based on a fixed memory structure $\mathcal{M}$. We proceed in three steps. First, in Section 4.3, we present the core concepts of this characterization, i.e., the properties that preference relations must satisfy to admit optimal strategies based on $\mathcal{M}$. Secondly, we state our equivalence result in Section 4.4, alongside the advertised one-to-two-player lift. We also provide an illustrative application of our characterization in Subsection 4.4.2.

4.3 Concepts

4.3.1 Generalizing monotony and selectivity

As explained in Section 3.1, Gimbert and Zielonka's characterization relies on notions of *monotony* and *selectivity* of the preference relation [GZ05].

[GZ05]: Gimbert et al. (2005), *Games Where You Can Play Optimally Without Any Memory*

Intuitively, the main difference between Gimbert and Zielonka's technical approach and ours is the following. In the memoryless setting, all the reasoning can be *abstracted away* from the underlying arena and done at the level of sequences of colors. In the finite-memory setting, however, one has to pay attention to how sequences of colors are composed and compared, to maintain consistency with regard to the memory and the underlying game arena. This required defining generalizations of monotony and selectivity *modulo a memory structure*.

> **Definition 4.3.1** ($\mathcal{M}$-monotony) *Let* $\mathcal{M} = (M, m_{\mathsf{init}}, \alpha_{\mathsf{upd}})$ *be a memory structure. A preference relation* $\sqsubseteq$ *is* $\mathcal{M}$-monotone *if for all* $m \in M$, *for all regular languages* K_1, K_2 *on alphabet* C,
>
> $$(\exists w \in L_{m_{\mathsf{init}}, m}, [w K_1] \sqsubset [w K_2])$$
> $$\implies (\forall w' \in L_{m_{\mathsf{init}}, m}, [w' K_1] \sqsubseteq [w' K_2]). \quad (4.7)$$

Note that $[w K_1] = w[K_1]$.

Intuitively, $\mathcal{M}$-*monotony* extends Gimbert and Zielonka's monotony by asking one to compare prefixes *belonging to the same language* $L_{m_{\mathsf{init}}, m}$, that

is, prefixes that are classified equivalently by the memory structure. This property roughly captures that $\sqsubseteq$ is *stable with regard to prefix addition*, for memory-equivalent prefixes.

The original monotony notion is exactly equivalent to our $\mathcal{M}$-monotony with $\mathcal{M}$ being the trivial memory structure $\mathcal{M}_{\text{triv}}$: that is, the memoryless case is naturally a specific subcase of our framework.

Example 4.3.2 We present a running example to illustrate our notions and the upcoming characterization: the conjunction of two reachability conditions, which is a subcase of *generalized reachability games*, studied extensively in [FH10]. Let $C = \{a, b, c\}$ be the set of colors. Formally, let the qualitative objective $W = \text{Reach}(a) \cap \text{Reach}(b)$ be the set of infinite words $w = c_1 c_2 \ldots$ such that

$$\exists i, j \in \mathbb{N}, c_i = a \wedge c_j = b.$$

This objective is ω-regular and is in particular recognized by the deterministic Büchi automaton in Figure 4.1 (left). We discussed how this qualitative objective W induces a preference relation after Definition 4.2.1.

We first show that this preference relation is not $\mathcal{M}_{\text{triv}}$-monotone (that is, is not *monotone* for [GZ05]). Take $K_1 = a^*$, $K_2 = b^*$. For $w = a$, $w' = b$, we have $[wK_1] \sqsubset [wK_2]$, but $[w'K_2] \sqsubset [w'K_1]$. This means that the preference relation is not stable with regard to prefix addition (at least, without distinguishing different classes of prefixes).

We exhibit a small memory structure $\mathcal{M}^p = (M^p, m_1^p, \alpha_{\text{upd}}^p)$ such that $\sqsubseteq$ is $\mathcal{M}^p$-monotone: it is pictured in Figure 4.1 (center).

Let us prove that $\sqsubseteq$ is $\mathcal{M}^p$-monotone. Let $m \in M^p$ and K_1, K_2 be regular languages; we want to show that Equation (4.7) is satisfied. We assume that there exists $w \in L_{m_1^p, m}$ such that $[wK_1] \sqsubset [wK_2]$: this means that all words of $[wK_1]$ are losing, and that there exists a winning word in $[wK_2]$. Let $w' \in L_{m_1^p, m}$; we show that we necessarily have that $[w'K_1] \sqsubseteq [w'K_2]$. Note that if $[K_1]$ is empty, this always holds; we now assume that $[K_1]$ is non-empty. We study the two possible values of m separately.

- If $m = m_1^p$, then w and w' do not see a. If w does not see b either, as there is a winning word in $[wK_2]$, then there must be a winning word in $[K_2]$. This word is still winning after prepending w' to it, so there is a winning word in $[w'K_2]$, and $[w'K_1] \sqsubseteq [w'K_2]$. If w sees b, then $[K_1]$ cannot have a word seeing a. As w' does not see a either, all words of $[w'K_1]$ are losing, so $[w'K_1] \sqsubseteq [w'K_2]$.
- If $m = m_2^p$, then w and w' see a. Clearly, w cannot see b (as $[wK_1]$ would contain a winning word). This implies that $[K_2]$ must contain a word reaching b; as w' reaches a, the concatenation of w' with the word of $[K_2]$ reaching b means that there is a winning word in $[w'K_2]$, so $[w'K_1] \sqsubseteq [w'K_2]$.

[FH10]: Fijalkow et al. (2010), *The surprizing complexity of reachability games*

This objective will also be a special case of a class of objectives studied in Chapter 7.

In general, we transfer properties defined on preference relations (such as $\mathcal{M}$-monotony) to qualitative objectives W by considering the preference relation $\sqsubseteq_W$ they induce.

[GZ05]: Gimbert et al. (2005), *Games Where You Can Play Optimally Without Any Memory*
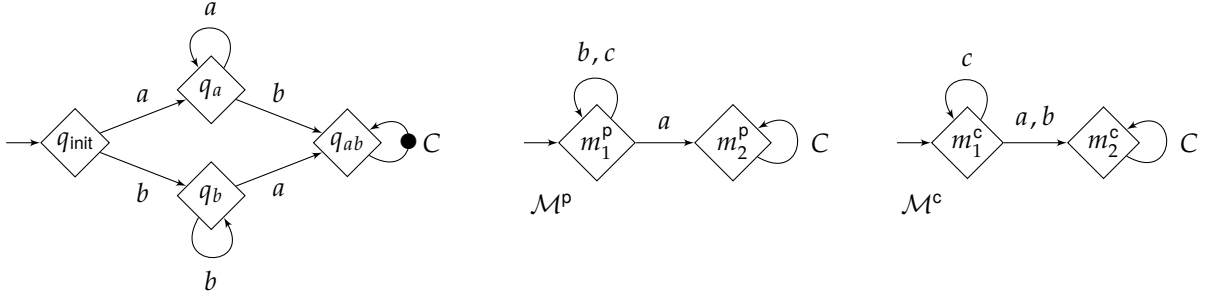
Figure 4.1: DBA recognizing $W = \mathsf{Reach}(a) \cap \mathsf{Reach}(b)$ (left), memory structures $\mathcal{M}^{\mathsf{p}}$ (center) and $\mathcal{M}^{\mathsf{c}}$ (right).

> **Remark 4.3.3** The $\mathcal{M}$-monotony notion is in general not *symmetric*, i.e.,
> it does not always hold for a preference relation if and only if it holds for
> its inverse. Let $C = \{a, b\}$ and let $x \in C^{\omega}$ be the non-ultimately-periodic
> word $abab^2ab^3\ldots$ (the argument can be adapted for any non-ultimately-
> periodic word). We consider the qualitative objective $W = \{x\}$.
> Objective W is not $\mathcal{M}_{\mathsf{triv}}$-monotone: if $w = \varepsilon$, $w' = a$, $K_1 = bC^*$, and
> $K_2 = aC^*$, we have $[wK_1] \sqsubset_W [wK_2]$, but $[w'K_2] \sqsubset_W [w'K_1]$.
> However, $\overline{W}$ is $\mathcal{M}_{\mathsf{triv}}$-monotone. Let K_1, K_2 be regular languages, and
> $w \in L_{m_{\mathsf{init}}, m_{\mathsf{init}}}$ be such that $[wK_1] \sqsubseteq_{\overline{W}} [wK_2]$. This implies that $[wK_1]$
> contains only words losing for $\overline{W}$ and that $[wK_2]$ contains a word in $\overline{W}$,
> i.e., an infinite word that is not x. As $[wK_2]$ is non-empty, it necessarily
> contains an ultimately periodic word: there is an infinite path in an NFA
> recognizing regular language wK_2, so there is an infinite path that loops
> around a cycle (this is essentially the same argument as in the proof of
> Lemma 2.7.12). Hence, for all $w' \in L_{m_{\mathsf{init}}, m_{\mathsf{init}}}$, the set $[w'K_2]$ also contains
> an ultimately periodic word, so a winning word for $\overline{W}$. We obtain that
> $[w'K_1] \sqsubseteq_{\overline{W}} [w'K_2]$.
> We will see that symmetry of $\mathcal{M}$-monotony holds for all ω-regular
> objectives in Subsection 4.3.2.

We now turn our focus to *selectivity*.

> **Definition 4.3.4** ($\mathcal{M}$-selectivity) *Let* $\mathcal{M} = (M, m_{\mathsf{init}}, \alpha_{\mathsf{upd}})$ *be a memory*
> *structure. A preference relation* $\sqsubseteq$ *is* $\mathcal{M}$-selective *if for all* $w \in C^*$, $m =$
> $\alpha^*_{\mathsf{upd}}(m_{\mathsf{init}}, w)$, *for all regular languages* K_1, K_2, K_3 *such that* $K_1, K_2 \subseteq L_{m,m}$,
>
> $$[w(K_1 \cup K_2)^*K_3] \sqsubseteq [wK_1^*] \cup [wK_2^*] \cup [wK_3]. \tag{4.8}$$

Similarly, $\mathcal{M}$-*selectivity* extends Gimbert and Zielonka's selectivity by
asking one to compare sequences of colors *belonging to the same language*
$L_{m,m}$, that is, sequences read as cycles on the memory structure. Note also
that the memory state m should be consistent with the prefix w read from
the initial memory state m_{init}. This property roughly captures that $\sqsubseteq$ is
stable with regard to cycle mixing, for memory-equivalent cycles. Again, the
original selectivity notion is equivalent to $\mathcal{M}_{\mathsf{triv}}$-selectivity.

Example 4.3.5 We go back to objective $W = \mathsf{Reach}(a) \cap \mathsf{Reach}(b)$, the monotony of which was discussed in Example 4.3.2.

First, objective W is not $\mathcal{M}_{\mathsf{triv}}$-selective: take w as the empty word, $K_1 = a^*$, $K_2 = b^*$, and $K_3 = c^*$; to win, K_1 and K_2 need to be "mixed". We exhibit a small memory structure $\mathcal{M}^{\mathsf{c}} = (M^{\mathsf{c}}, m_1^{\mathsf{c}}, \alpha_{\mathsf{upd}}^{\mathsf{c}})$ such that $\sqsubseteq$ is $\mathcal{M}^{\mathsf{c}}$-selective: it is pictured in Figure 4.1 (right). We prove that $\sqsubseteq$ is $\mathcal{M}^{\mathsf{c}}$-selective. Let $w \in C^*$, $m = (\alpha_{\mathsf{upd}}^{\mathsf{c}})^*(m_1^{\mathsf{c}}, w)$, and K_1, K_2, K_3 be regular languages such that $K_1, K_2 \subseteq L_{m,m}$. We show that Equation (4.8) is satisfied, i.e., that

$$[w(K_1 \cup K_2)^* K_3] \sqsubseteq [wK_1^*] \cup [wK_2^*] \cup [wK_3].$$

If all words of $[w(K_1 \cup K_2)^* K_3]$ are losing, this equation trivially holds; we thus assume that this set contains a winning word. We therefore have to show that there is a winning word in $[wK_1^*]$, $[wK_2^*]$, or $[wK_3]$. We study the two possible values of m separately.

- ▶ If $m = m_1^{\mathsf{c}}$, then w does not reach a nor b, and the same holds for all words of K_1 and K_2, as $K_1, K_2 \subseteq L_{m,m}$. Therefore, if a word of $[w(K_1 \cup K_2)^* K_3]$ is winning, this must be because a word of $[wK_3]$ is winning.
- ▶ If $m = m_2^{\mathsf{c}}$, we distinguish three cases. If w reaches both a and b, then $[wK_1^*] \cup [wK_2^*] \cup [wK_3]$ trivially contains only winning words. If w reaches a but not b, then there must be a word reaching b in $[(K_1 \cup K_2)^* K_3]$. Hence, at least one set among $[K_1^*]$, $[K_2^*]$, and $[K_3]$ must contain a word reaching b, so $[wK_1^*]$, $[wK_2^*]$, or $[wK_3]$ contains a winning word. A symmetric argument works if w reaches b but not a.

We can also show that $\overline{W}$ is $\mathcal{M}_{\mathsf{triv}}$-selective using similar arguments.

In a nutshell, $\mathcal{M}$-monotony deals with prefixes up to the first cycle (on the memory structure) and $\mathcal{M}$-selectivity deals with the cycles thereafter; we will see that memory structures can be built in a compositional way based on these two complementary tasks.

Our notions respect the natural intuition that access to additional memory should always be helpful: if a memory structure $\mathcal{M}$ is sufficient to classify sequences of colors in a way that guarantees $\mathcal{M}$-monotony or $\mathcal{M}$-selectivity, then it should also be the case for "more powerful" structures.

Lemma 4.3.6 *Let $\mathcal{M}$ and $\mathcal{M}'$ be two memory structures. If $\sqsubseteq$ is $\mathcal{M}$-monotone (resp. $\mathcal{M}$-selective) then, it is also $(\mathcal{M} \otimes \mathcal{M}')$-monotone (resp. $(\mathcal{M} \otimes \mathcal{M}')$-selective).*

Proof. We write $\mathcal{M} = (M, m_{\mathsf{init}}, \alpha_{\mathsf{upd}})$ and $\mathcal{M}' = (M', m'_{\mathsf{init}}, \alpha'_{\mathsf{upd}})$.

Let us assume that $\sqsubseteq$ is $\mathcal{M}$-monotone, that is, for all $m \in M$, for all regular

languages K_1, K_2,

$$\exists w \in L_{m_{\text{init}}, m}, [wK_1] \sqsubset [wK_2]$$
$$\implies \forall w' \in L_{m_{\text{init}}, m}, [w'K_1] \sqsubseteq [w'K_2]. \quad (4.9)$$

We show that $\sqsubseteq$ is $(\mathcal{M} \otimes \mathcal{M}')$-monotone, that is, for all $(m, m') \in M \times M'$, for all regular languages K_1, K_2,

$$\exists w \in L_{(m_{\text{init}}, m'_{\text{init}}),(m,m')}, [wK_1] \sqsubset [wK_2]$$
$$\implies \forall w' \in L_{(m_{\text{init}}, m'_{\text{init}}),(m,m')}, [w'K_1] \sqsubseteq [w'K_2]. \quad (4.10)$$

To do so, we notice that $L_{(m_{\text{init}}, m'_{\text{init}}),(m,m')} \subseteq L_{m_{\text{init}}, m}$ (the product of memory structures simply updates both memories in parallel). Thus, if the premise of Equation (4.10) holds, we obtain by Equation (4.9) that the conclusion of Equation (4.10) also holds.

A similar argument can be laid out to show that $\mathcal{M}$-selectivity implies $(\mathcal{M} \otimes \mathcal{M}')$-selectivity. It is enough to notice that for all $(m, m') \in M \times M'$, we have $L_{(m,m'),(m,m')} \subseteq L_{m,m}$: the definition of $\mathcal{M}$-selectivity is thus clearly stronger than the definition of $(\mathcal{M} \otimes \mathcal{M}')$-selectivity. $\qquad \square$

4.3.2 Discussion about the $\mathcal{M}$-monotony notion

We discuss the significance of the restriction of K_1 and K_2 to *regular* languages in the definition of $\mathcal{M}$-monotony. Observations in this section will hopefully bring insight to the reader, and will be revisited and find additional meaning in the subsequent Chapter 7. In particular, we show that $\mathcal{M}$-monotony always coincides for a qualitative objective and for its complement in the ω-regular case, but not in general. This will simplify the multiple studies of ω-regular examples later on.

Let $W \subseteq C^\omega$ be a qualitative objective. We recall that the prefix preorder $\leq$ of W compares finite words w.r.t. inclusion of their winning continuations. The $\mathcal{M}$-monotony notion also compares in some way finite words with respect to their continuations using infinite extensions of *regular* languages.

See Section 2.8 for the formal definition of prefix preorder of an objective.

We discuss what happens if we drop this regular requirement by defining *$\mathcal{M}$-strong-monotony*. A similar notion called *strong monotony* has already been defined [BFMM11] and turns out to be equivalent to our notion of $\mathcal{M}_{\text{triv}}$-strong-monotony. We can therefore see once again our definition as a reformulation and a generalization to handle arbitrary memory structures, rather than only the "memoryless memory structure" $\mathcal{M}_{\text{triv}}$.

[BFMM11]: Bianco et al. (2011), *Exploring the boundary of half-positionality*

Definition 4.3.7 ($\mathcal{M}$-strong-monotony) *Let* $\mathcal{M} = (M, m_{\text{init}}, \alpha_{\text{upd}})$ *be a memory structure. A preference relation* $\sqsubseteq$ *is* $\mathcal{M}$-strongly-monotone *if for all*

$m \in M$, for all sets $K_1, K_2 \subseteq C^\omega$,

$$(\exists w \in L_{m_{\text{init}}, m}, w K_1 \sqsubset w K_2) \implies (\forall w' \in L_{m_{\text{init}}, m}, w' K_1 \sqsubseteq w' K_2).$$

For qualitative objectives, we can reformulate this definition to be about comparing *prefixes*: the original definition compares continuations with each other by looking at what prefixes make them winning; the following reformulation compares instead prefixes with each other by looking at what continuations make them winning, through the prefix preorder $\preceq$.

Lemma 4.3.8 *Let $W \subseteq C^\omega$ be an objective and $\mathcal{M} = (M, m_{\text{init}}, \alpha_{\text{upd}})$ be a memory structure. Objective W is $\mathcal{M}$-strongly-monotone if and only if for all $m \in M$, for all $w, w' \in L_{m_{\text{init}}, m}$, w and w' are comparable for the prefix preorder $\preceq$ of W.*

Proof. We first prove the implication from left to right by contraposition. Let $m \in M$ and $w, w' \in L_{m_{\text{init}}, m}$. If w and w' are incomparable for $\preceq$, it means that there are $x_1, x_2 \in C^\omega$ such that

$$w x_1 \in W, \quad w x_2 \notin W$$
$$w' x_1 \notin W, \quad w' x_2 \in W.$$

If we take $K_1 = \{x_1\}$ and $K_2 = \{x_2\}$, then $w K_2 \sqsubset w K_1$ and $w' K_1 \sqsubset w' K_2$, so W is not $\mathcal{M}$-strongly-monotone.

We now prove the implication from right to left directly. We assume that for all $m \in M$, for all $w, w' \in L_{m_{\text{init}}, m}$, w and w' are comparable for $\preceq$. Taking the definition of $\mathcal{M}$-strong-monotony, let $m \in M$, $K_1, K_2 \subseteq C^\omega$, and $w \in L_{m_{\text{init}}, m}$ such that $w K_1 \sqsubset w K_2$. This means that there is no winning word in $w K_1$ and that there is a winning word in $w K_2$.

Now, let $w' \in L_{m_{\text{init}}, m}$. We want to show that $w' K_1 \sqsubseteq w' K_2$, i.e., that $w' K_1$ contains no winning word or that $w' K_2$ contains a winning word. We have by hypothesis that w is comparable to w' for $\preceq$. If $w \preceq w'$, then $w' K_2$ also contains a winning word, which ends the proof. If $w' \preceq w$, then $w' K_1$ also contains no winning word, which also ends the proof. $\qquad\square$

We discuss a few differences between $\mathcal{M}$-monotony and $\mathcal{M}$-strong-monotony. First, as opposed to $\mathcal{M}$-monotony (Remark 4.3.3), $\mathcal{M}$-strong-monotony holds for an objective if and only if it holds for its complement, which is easily proved using the previous lemma.

Lemma 4.3.9 *Let $\mathcal{M}$ be a memory structure. An objective is $\mathcal{M}$-strongly-monotone if and only if its complement is.*

Proof. We write $\mathcal{M} = (M, m_{\text{init}}, \alpha_{\text{upd}})$. Let $W \subseteq C^\omega$ be an $\mathcal{M}$-strongly-monotone objective. We use the reformulation of $\mathcal{M}$-strong-monotony

provided by Lemma 4.3.8. Let $m \in M$ and $w, w' \in L_{m_{\text{init}}, m}$. Then, w and w' are comparable for $\preceq_W$. As $\preceq_{\overline{W}} = \succeq_W$, w and w' are also comparable for $\preceq_{\overline{W}}$. Hence, $\overline{W}$ is $\mathcal{M}$-strongly-monotone too. □

The $\mathcal{M}$-strong-monotony notion is indeed stronger in general, but both notions coincide for ω-regular objectives. This is the object of the next two lemmas.

> **Lemma 4.3.10** *There exists an $\mathcal{M}_{\text{triv}}$-monotone objective that is not $\mathcal{M}_{\text{triv}}$-strongly-monotone.*

Proof. In Remark 4.3.3, we have seen an objective W that is not $\mathcal{M}_{\text{triv}}$-monotone but whose complement $\overline{W}$ is $\mathcal{M}_{\text{triv}}$-monotone. As $\mathcal{M}_{\text{triv}}$-strong-monotony is stronger than $\mathcal{M}_{\text{triv}}$-monotony, W is not $\mathcal{M}_{\text{triv}}$-strongly-monotone. By the above Lemma 4.3.9, $\mathcal{M}_{\text{triv}}$-strong-monotony is symmetric, so $\overline{W}$ is not $\mathcal{M}_{\text{triv}}$-strongly-monotone either. Therefore, $\overline{W}$ is $\mathcal{M}$-monotone but not $\mathcal{M}$-strongly-monotone. □

> **Lemma 4.3.11** *Let $\mathcal{M}$ be a memory structure. An ω-regular objective is $\mathcal{M}$-monotone if and only if it is $\mathcal{M}$-strongly-monotone. In particular, an ω-regular objective is $\mathcal{M}$-monotone if and only if its complement is.*

Proof. Let $W \subseteq C^\omega$ be an ω-regular objective. The implication from $\mathcal{M}$-strong-monotony to $\mathcal{M}$-monotony is obvious: $\mathcal{M}$-monotony quantifies universally over fewer languages.

Assume now that W is $\mathcal{M}$-monotone. We write $\mathcal{M} = (M, m_{\text{init}}, \alpha_{\text{upd}})$. We show that W is $\mathcal{M}$-strongly-monotone by using the reformulation of Lemma 4.3.8. Let $m \in M$ and $w, w' \in L_{m_{\text{init}}, m}$. We show that w and w' are comparable for $\preceq$. We assume that $w \not\preceq w'$ and we show that $w' \preceq w$, i.e., that $(w')^{-1}W \subseteq w^{-1}W$. As $(w')^{-1}W$ and $w^{-1}W$ are ω-regular, it suffices to show that all ultimately periodic words in $(w')^{-1}W$ are in $w^{-1}W$ (using Lemma 2.7.12 about ultimately periodic separation of distinct ω-regular objectives). Let $x' \in C^*$, $y' \in C^+$ such that $x'(y')^\omega \in (w')^{-1}W$; we show that $x'(y')^\omega \in w^{-1}W$. As $w \not\preceq w'$, also by Lemma 2.7.12, there is $x \in C^*$, $y \in C^+$ such that $xy^\omega \in w^{-1}W$ but $xy^\omega \notin (w')^{-1}W$. We take $K_1 = xy^*$ and $K_2 = x'(y')^*$ (which are regular languages). We have $[K_1] = \{xy^\omega\}$ and $[K_2] = \{x'(y')^\omega\}$. Therefore, $[w'K_1] = \{w'xy^\omega\} \sqsubset \{w'x'(y')^\omega\} = w'[K_2]$. By $\mathcal{M}$-monotony of W and as $w, w' \in L_{m_{\text{init}}, m}$, $[wK_1] \sqsubseteq [wK_2]$. As $[wK_1] = \{wxy^\omega\}$ contains a winning word, so does $[wK_2] = \{wx'(y')^\omega\}$. Hence, $wx'(y')^\omega \in W$, or in other words, $x'(y')^\omega \in w^{-1}W$, which is what we wanted to show.

For the second claim of the lemma, we have that W is $\mathcal{M}$-monotone if and only if W is $\mathcal{M}$-strongly-monotone (first claim) if and only if $\overline{W}$ is $\mathcal{M}$-strongly-monotone (Lemma 4.3.9) if and only if $\overline{W}$ is $\mathcal{M}$-monotone. □

We also have for similar reasons that for an ω-regular objective W, W is $\mathcal{M}$-selective if and only if it is "$\mathcal{M}$-selective without restricting K_1, K_2, and K_3 to being regular languages". However, this observation did not seem to bring much insight (unlike for $\mathcal{M}$-monotony, for which we have a useful reformulation in Lemma 4.3.8) and will not be needed elsewhere in this thesis.

4.3.3 Prefix-covers and cyclic-covers

While the aforementioned concepts of $\mathcal{M}$-monotony and $\mathcal{M}$-selectivity are the primordial ones for stating the characterization, we still need two additional notions to prove it.

To prove that monotone and selective preference relations yield memoryless optimal strategies, Gimbert and Zielonka deploy an *inductive argument* on the number of choices in finite arenas. Intuitively, we want to use a similar approach for optimal strategies based on a memory structure, but because of the coupling between the memory structure and the finite arena (e.g., Lemma 4.2.9), the induction argument breaks, as adding one choice in the arena results in adding many in the *product arena* (as many as there are memory states), where the reasoning needs to occur.

To solve this issue, *we decouple the two aspects* (see Section 4.6). For a given memory structure $\mathcal{M}$, we first establish that on arenas that inherently share the same good properties as product arenas with $\mathcal{M}$ (that is, they already "classify" prefixes and cycles as $\mathcal{M}$ would), we can deploy the induction argument and obtain memoryless optimal strategies. Then, we obtain the result for optimal strategies based on $\mathcal{M}$ on arbitrary arenas as a corollary. The crux is identifying such "good" arenas: this is done through the following notions.

> **Definition 4.3.12** (Prefix-cover, cyclic-cover) *Let* $\mathcal{M} = (M, m_{\mathsf{init}}, \alpha_{\mathsf{upd}})$ *be a memory structure and* $\mathcal{A} = (V, V_1, V_2, E)$ *be an arena. Let* $V_{\mathsf{cov}} \subseteq V$.
> *We say that* $\mathcal{M}$ *is a* prefix-cover *of* V_{cov} *in* $\mathcal{A}$ *if for all* $v \in V$, *there exists* $m_v \in M$ *such that, for all* $\gamma \in \mathsf{Hists}(\mathcal{A})$ *such that* $\mathsf{in}(\gamma) \in V_{\mathsf{cov}}$, $\mathsf{out}(\gamma) = v$ *and such that for all proper prefixes* γ' *of* γ, $\mathsf{out}(\gamma') \neq v$, *we have* $\alpha_{\mathsf{upd}}^*(m_{\mathsf{init}}, \mathsf{col}^*(\gamma)) = m_v$.
> *We say that* $\mathcal{M}$ *is a* cyclic-cover *of* V_{cov} *in* $\mathcal{A}$ *if for all* $\gamma \in \mathsf{Hists}(\mathcal{A})$ *such that* $\mathsf{in}(\gamma) \in V_{\mathsf{cov}}$, *if* $v = \mathsf{out}(\gamma)$ *and* $m = \alpha_{\mathsf{upd}}^*(m_{\mathsf{init}}, \mathsf{col}^*(\gamma))$, *for all* $\gamma' \in \mathsf{Hists}(\mathcal{A})$ *such that* $\mathsf{in}(\gamma') = \mathsf{out}(\gamma') = v$, $\alpha_{\mathsf{upd}}^*(m, \mathsf{col}^*(\gamma')) = m$.

Intuitively, $\mathcal{M}$ is a prefix-cover of a set of vertices V_{cov} if the histories starting in V_{cov} and visiting a given vertex $v \in V$ for the first time are read up to the same memory state in the memory structure. Similarly, $\mathcal{M}$ is a cyclic-cover of $\mathcal{A}$ if the cycles of $\mathcal{A}$ are read as cycles in the memory structure, once the memory has been initialized properly.

Definition 4.3.12 can be equivalently stated by considering simple cycles only.

As hinted above, the canonical example of a prefix-covered and cyclic-covered arena is a product arena.

> **Lemma 4.3.13** *Let* $\mathcal{M} = (M, m_{\mathsf{init}}, \alpha_{\mathsf{upd}})$ *be a memory structure and* $\mathcal{A} = (V, V_1, V_2, E)$ *be an arena. Structure* $\mathcal{M}$ *is both a prefix-cover and a cyclic-cover of* $V_{\mathsf{cov}} = V \times \{m_{\mathsf{init}}\}$ *in the product arena* $\mathcal{A} \ltimes \mathcal{M}$.
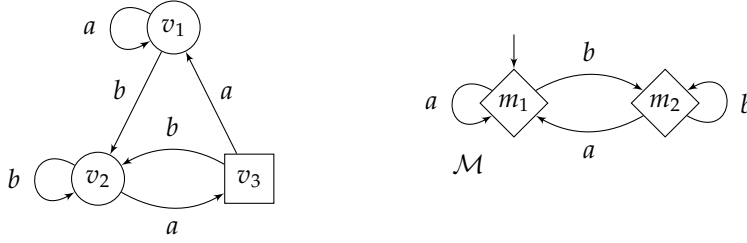
Figure 4.2: Memory structure $\mathcal{M}$ (right) is a prefix-cover and cyclic-cover of $\{v_1, v_3\}$ in the arena on the left, but the arena cannot be realized as is as the product of an arena with $\mathcal{M}$.

Proof. The main argument in this proof is that if there is a history γ of the product arena $\mathcal{A} \ltimes \mathcal{M}$ with $\mathsf{in}(\gamma) = (v, m)$ and $\mathsf{out}(\gamma) = (v', m')$, then reading $\mathsf{col}^*(\gamma)$ from m in the memory structure $\mathcal{M}$ leads to m' (i.e., $\alpha^*_{\mathsf{upd}}(m, \mathsf{col}^*(\gamma)) = m'$). This can be proved by induction on the length of γ, thanks to how the product arena is built.

We first show that $\mathcal{M}$ is a prefix-cover of $V_{\mathsf{cov}} = V \times \{m_{\mathsf{init}}\}$ in the product arena $\mathcal{A} \ltimes \mathcal{M}$. What we have to prove, instantiating the definition of prefix-cover in this case, is that for all $(v, m) \in V \times M$, there exists $m_{(v,m)} \in M$ such that, for all $\gamma \in \mathsf{Hists}(\mathcal{A} \ltimes \mathcal{M})$ such that $\mathsf{in}(\gamma) \in V_{\mathsf{cov}}$, $\mathsf{out}(\gamma) = (v, m)$ and such that for all γ' proper prefix of γ, $\mathsf{out}(\gamma') \neq v$, we have $\alpha^*_{\mathsf{upd}}(m_{\mathsf{init}}, \mathsf{col}^*(\gamma)) = m_{(v,m)}$. Let $(v, m) \in V \times M$; we take $m_{(v,m)} = m$. Then, if $\gamma \in \mathsf{Hists}(\mathcal{A} \ltimes \mathcal{M})$ is such that $\mathsf{in}(\gamma) \in V_{\mathsf{cov}}$ (that is, is equal to (v', m_{init}) for some $v' \in V$), and $\mathsf{out}(\gamma) = (v, m)$, we have by construction of the product arena that $\alpha^*_{\mathsf{upd}}(m_{\mathsf{init}}, \mathsf{col}^*(\gamma)) = m = m_{(v,m)}$, as required.

To prove that $\mathcal{M}$ is a cyclic-cover of V_{cov} in $\mathcal{A} \ltimes \mathcal{M}$, we have to prove that for all $\gamma \in \mathsf{Hists}(\mathcal{A} \ltimes \mathcal{M})$ such that $\mathsf{in}(\gamma) \in V_{\mathsf{cov}}$, if $(v, m) = \mathsf{out}(\gamma)$ and $m' = \alpha^*_{\mathsf{upd}}(m_{\mathsf{init}}, \mathsf{col}^*(\gamma))$, for all $\gamma' \in \mathsf{Hists}(\mathcal{A} \ltimes \mathcal{M})$ such that $\mathsf{in}(\gamma') = \mathsf{out}(\gamma') = (v, m)$, $\alpha^*_{\mathsf{upd}}(m', \mathsf{col}^*(\gamma')) = m'$. Let $\gamma \in \mathsf{Hists}(\mathcal{A} \ltimes \mathcal{M})$ such that $\mathsf{in}(\gamma) \in V_{\mathsf{cov}}$ (that is, $\mathsf{in}(\gamma) = (v', m_{\mathsf{init}})$ for some $v' \in V$). Then, if $(v, m) = \mathsf{out}(\gamma)$, we have by construction of the product arena that $m' = \alpha^*_{\mathsf{upd}}(m_{\mathsf{init}}, \mathsf{col}^*(\gamma)) = m$. Let $\gamma' \in \mathsf{Hists}(\mathcal{A} \ltimes \mathcal{M})$ such that $\mathsf{in}(\gamma') = \mathsf{out}(\gamma') = (v, m)$. By construction of the product arena, we therefore have that $\alpha^*_{\mathsf{upd}}(m, \mathsf{col}^*(\gamma')) = m$, as required. $\square$

Still, not only product arenas are prefix-covered and cyclic-covered.

Example 4.3.14 We consider the arena in Figure 4.2 (left). Memory structure $\mathcal{M}$ (right) is both a prefix-cover and a cyclic-cover of $\{v_1, v_3\}$ in this arena. This arena cannot be realized as a product of a smaller arena with $\mathcal{M}$; in particular, there would need to be at least two copies of the vertex controlled by $\mathcal{P}_2$.

Remark 4.3.15 We discovered after defining these covering notions that something similar had already been defined in an unpublished part of Kopczyński's thesis [Kop08, Definition 8.12]. Our notion of an arena being both prefix-covered and cyclic-covered by a memory structure $\mathcal{M}$ is equivalent to his notion that an arena *adheres to memory $\mathcal{M}$*,

[Kop08]: Kopczyński (2008), *Half-positional Determinacy of Infinite Games*

which means that it is possible to assign a state of $\mathcal{M}$ to every vertex of the arena such that moving along the edges of the arena updates these memory states in a consistent way. Let $\mathcal{A} = (V, V_1, V_2, E)$ and $\mathcal{M} = (M, m_{\text{init}}, \alpha_{\text{upd}})$. We say that $\mathcal{A}$ *adheres to* $\mathcal{M}$ *from* $V_{\text{cov}} \subseteq V$ if there exists $\phi \colon V \to M$ such that $\phi(V_{\text{cov}}) \subseteq \{m_{\text{init}}\}$ and for all $(v, c, v') \in E$, $\alpha_{\text{upd}}(\phi(v), c) = \phi(v')$. In other words, an arena adheres to a memory structure if there is a morphism from the arena vertices to the memory states, while preserving some distinguished initial states.

Our definitions of *prefix-cover* and *cyclic-cover* can be seen as two distinct sides of this idea of *adherence* which, when put together, are actually equivalent to it. We distinguish in every technical lemma which kind of covering we actually need, but our general scheme would work by just using the more straightforward notion of *adherence* instead.

In Kopczyński's thesis, all objectives are prefix-independent, and this entails that distinguishing the initial state of the memory is less crucial [Kop08, Proposition 8.2]. We therefore slightly adapted his definition of *adherence* to fit our more general setting by distinguishing initial states in the morphism.

4.4 Characterization

4.4.1 Main results

We now have the necessary ingredients to state our general equivalence result formally. We defer the formal proofs of both directions of the equivalence to Sections 4.5 and 4.6, and only explain here how to combine them.

Theorem 4.4.1 (Characterization of AIFM determinacy over finite arenas)
Let $\sqsubseteq$ be a preference relation and let $\mathcal{M}$ be a memory structure. Then, both players have optimal strategies based on memory structure $\mathcal{M}$ in finite arenas if and only if $\sqsubseteq$ and $\sqsubseteq^{-1}$ are $\mathcal{M}$-monotone and $\mathcal{M}$-selective.

This result is meant to mirror the result of Gimbert and Zielonka [GZ05, Theorem 2]: their result can be retrieved from Theorem 4.4.1 by taking the trivial memory structure $\mathcal{M}_{\text{triv}}$. As such, our work brings a strict generalization of Gimbert and Zielonka's results [GZ05] to the finite-memory case.

[GZ05]: Gimbert et al. (2005), *Games Where You Can Play Optimally Without Any Memory*

Remark 4.4.2 The statements of our intermediate results are slightly stronger than what is stated in the summarizing characterization above. Although we only need optimal strategies in the left-to-right direction (Propositions 4.5.1 and 4.5.2), we prove the stronger existence of finite-memory Nash equilibria in the other direction (Proposition 4.6.3 and Corollary 4.6.5). In particular, the existence of arena-independent finite-memory optimal strategies in all finite arenas entails determinacy of the preference relation (see Remark 4.2.7) over finite arenas.

Similarly, we study the two implications of the equivalence in a *compositional* way: we split the reasoning for $\mathcal{M}$-monotony and $\mathcal{M}$-selectivity, using different memory structures for each whenever meaningful, as

well as for the players.

To prove Theorem 4.4.1, we invoke the results we will prove in Sections 4.5 and 4.6.

Proof of Theorem 4.4.1. The left-to-right implication follows from Proposition 4.5.1 (for $\mathcal{M}$-monotony) and Proposition 4.5.2 (for $\mathcal{M}$-selectivity), applied to each player with respect to its preference relation. The converse implication is established in Corollary 4.6.5 about Nash equilibria, which can be restated in terms of optimal strategies through Lemma 4.2.6. □

As a by-product of our method, we also obtain a similar equivalence by solely considering one of the two players and their corresponding one-player arenas.

> **Theorem 4.4.3** (One-player characterization) *Let* $\sqsubseteq$ *be a preference relation and* $\mathcal{M}$ *be a memory structure. Then,* $\mathcal{P}_1$ *has optimal strategies based on* $\mathcal{M}$ *in all its finite one-player arenas if and only if* $\sqsubseteq$ *is* $\mathcal{M}$*-monotone and* $\mathcal{M}$*-selective.*

Although this looks like a weak version of Theorem 4.4.1 at first sight, this is actually a distinct result as both sides of the equivalence are weaker: on the left side, it only handles the memory requirements for the one-player games of $\mathcal{P}_1$; on the right side, it does not assume anything about the inverse preference relation $\sqsubseteq^{-1}$.

Albeit close, this is also distinct from the *half-positional determinacy* result from [Kop06, Theorem 4], which gives *sufficient* conditions about an objective for a player to admit memoryless optimal strategies on every finite *two-player* arena — in Theorem 4.4.3, we give a *necessary and sufficient* condition for a player to admit arena-independent finite-memory optimal strategies on its finite *one-player* arenas only. The sufficient conditions from [Kop06] (*prefix-independence* and *concavity*) imply $\mathcal{M}_{\mathsf{triv}}$-monotony and $\mathcal{M}_{\mathsf{triv}}$-selectivity, but not the other way around. Given a preference relation, it is possible for a player to have arena-independent finite-memory optimal strategies on its finite one-player arenas, but not on all finite two-player arenas; see, e.g., the example used in [Kop06, Proposition 2]. In such an example, Theorem 4.4.3 could be applied, but not the result from [Kop06].

[Kop06]: Kopczyński (2006), *Half-Positional Determinacy of Infinite Games*

Proof of Theorem 4.4.3. The left-to-right implication follows from Proposition 4.5.1 (for $\mathcal{M}$-monotony) and Proposition 4.5.2 (for $\mathcal{M}$-selectivity). The converse implication is established in Corollary 4.6.6. □

One-to-two-player lift. Alongside the aforementioned equivalence result, Gimbert and Zielonka provide a corollary of high practical interest [GZ05, Corollary 7]: they essentially obtain as a by-product of their approach that if memoryless strategies suffice in all finite one-player

[GZ05]: Gimbert et al. (2005), *Games Where You Can Play Optimally Without Any Memory*

games of $\mathcal{P}_1$ and all finite one-player games of $\mathcal{P}_2$, they also suffice in all finite two-player games.

We are able to lift this corollary to the arena-independent finite-memory case, as follows.

Theorem 4.4.4 (One-to-two-player lift, finite arenas) *Let $\sqsubseteq$ be a preference relation and $\mathcal{M}_1$, $\mathcal{M}_2$ be two memory structures. Assume that*

1. *for all games played on a finite one-player arena of $\mathcal{P}_1$, $\mathcal{P}_1$ has an optimal strategy based on $\mathcal{M}_1$;*
2. *for all games played on a finite one-player arena of $\mathcal{P}_2$, $\mathcal{P}_2$ has an optimal strategy based on $\mathcal{M}_2$.*

Then, in all games played on finite (two-player) arenas, both $\mathcal{P}_1$ and $\mathcal{P}_2$ have optimal strategies based on memory structure $\mathcal{M} = \mathcal{M}_1 \otimes \mathcal{M}_2$.

We highlight the two (possibly different) memory structures of the two players to maintain a compositional approach, but if the same memory structure $\mathcal{M}$ works in both one-player versions, it also suffices in the two-player version (as $\mathcal{M} \otimes \mathcal{M}$ is isomorphic to $\mathcal{M}$).

Proof of Theorem 4.4.4. By Propositions 4.5.1 and 4.5.2, which essentially state that the left-to-right implication of Theorem 4.4.1 holds already in finite one-player games, the hypothesis yields that $\sqsubseteq$ is $\mathcal{M}_1$-monotone and $\mathcal{M}_1$-selective, while $\sqsubseteq^{-1}$ is $\mathcal{M}_2$-monotone and $\mathcal{M}_2$-selective. Now it suffices to apply Corollary 4.6.5 — essentially the right-to-left implication of Theorem 4.4.1 — to get the claim. □

4.4.2 Running example

We go back to our running example $W = \mathsf{Reach}(a) \cap \mathsf{Reach}(b)$, the monotony of which was studied in Example 4.3.2 and the selectivity of which was studied in Example 4.3.5. We use Theorem 4.4.1 directly in order to illustrate it. In practice, using Theorem 4.4.4 may be preferable, as it yields a shorter proof: by exhibiting the right memory structure for $\mathcal{P}_1$ and $\mathcal{P}_2$, we simply have to show that these memory structures are sufficient to play optimally on both players' finite one-player arenas. But here, we already have detailed information on monotony and selectivity of W.

We discussed that W is $\mathcal{M}^\mathsf{p}$-monotone and $\mathcal{M}^\mathsf{c}$-selective ($\mathcal{M}^\mathsf{p}$ and $\mathcal{M}^\mathsf{c}$ are depicted in Figure 4.1), and that $\overline{W}$ is $\mathcal{M}_\mathsf{triv}$-selective. As W is ω-regular, its complement $\overline{W}$ is also $\mathcal{M}^\mathsf{p}$-monotone (Lemma 4.3.11). Let $\mathcal{M} = \mathcal{M}^\mathsf{p} \otimes \mathcal{M}^\mathsf{c} \otimes \mathcal{M}_\mathsf{triv}$ be the product of all the considered memory structures. Structure $\mathcal{M}$ is drawn in Figure 4.3 (center). Although $\mathcal{M}$ formally has four states, only three of them are reachable from the initial state. By Lemma 4.3.6, we have that both $\sqsubseteq$ and $\sqsubseteq^{-1}$ are $\mathcal{M}$-monotone and
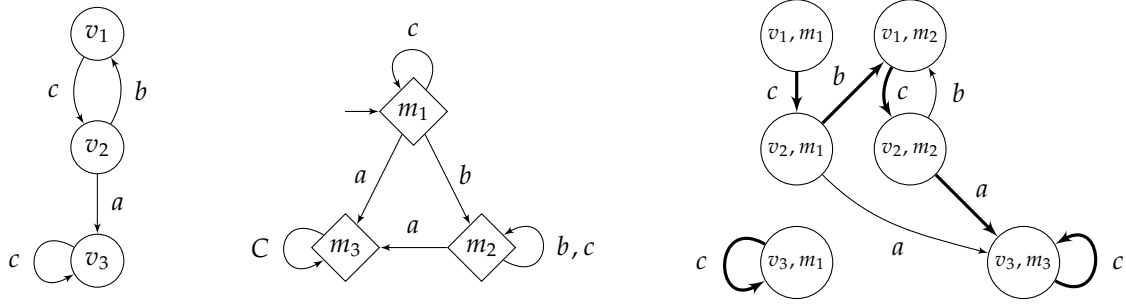
Figure 4.3: Arena $\mathcal{A}$ (left), memory structure $\mathcal{M}$ (center; with $m_{\text{init}} = m_1$), and product arena $\mathcal{A} \ltimes \mathcal{M}$ (right; only vertices reachable from $V \times \{m_{\text{init}}\}$ are depicted). The memoryless optimal strategy from $V \times \{m_{\text{init}}\}$ is highlighted with bold arrows.

$\mathcal{M}$-selective. Using Theorem 4.4.1, we obtain that *both players have optimal strategies based on memory structure $\mathcal{M}$ in all games played on finite arenas*. Note that memory structure $\mathcal{M}$ is minimal (no memory structure with fewer states suffices for $\mathcal{P}_1$ to play optimally in all arenas [FH10]).

We provide an example of a one-player arena $\mathcal{A} = (V, V_1, V_2 = \emptyset, E)$ in Figure 4.3, and show that there is an optimal strategy based on memory structure $\mathcal{M}$ for the preference relation $\sqsubseteq$. To do so, we invoke Lemma 4.2.9: we show equivalently that the product $\mathcal{A} \ltimes \mathcal{M}$ admits a memoryless optimal strategy from $V \times \{m_{\text{init}}\}$ for $\sqsubseteq$. Notice that no memoryless strategy suffices to play optimally in $(\mathcal{A}, W)$, as when starting in v_2, $\mathcal{P}_1$ should first visit v_1 before going to v_3. Also, the memoryless optimal strategy from $V \times \{m_{\text{init}}\}$ in the product arena is only optimal if the initial vertex is in $V \times \{m_{\text{init}}\}$; it is for instance not as good as possible from vertex (v_2, m_2).

4.5 From strategies based on $\mathcal{M}$ to $\mathcal{M}$-monotony and $\mathcal{M}$-selectivity

In this section, we show the necessity of $\mathcal{M}$-monotony and $\mathcal{M}$-selectivity for the existence of optimal strategies based on $\mathcal{M}$ in finite one-player arenas.

Monotony. We start with $\mathcal{M}$-monotony.

Proposition 4.5.1 (Necessity of $\mathcal{M}$-monotony) *Let $\sqsubseteq$ be a preference relation and $\mathcal{M}$ be a memory structure. If $\mathcal{P}_1$ has an optimal strategy based on $\mathcal{M}$ in all its finite one-player arenas, then $\sqsubseteq$ is $\mathcal{M}$-monotone.*

Note that this result can be instantiated for $\mathcal{P}_2$ and $\sqsubseteq^{-1}$ symmetrically.

Our proof can be sketched as follows. We need to establish that Equation (4.7) holds. We first instantiate the four languages involved in it: $\{w\}$, $\{w'\}$, K_1 and K_2. We take NFAs recognizing them and build an NFA $\mathcal{N}$ that joins them in such a way that, when $\mathcal{N}$ is considered as a game arena
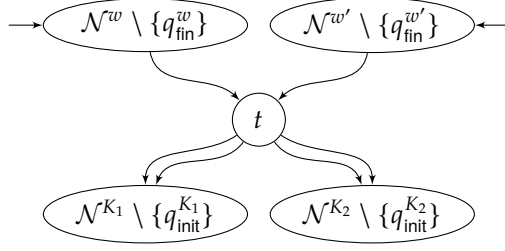
Figure 4.4: NFA $\mathcal{N}$ built to establish $\mathcal{M}$-monotony.

(see Lemma 4.2.12), its plays correspond exactly to the languages of infinite words considered in Equation (4.7). This arena is essentially composed of two chains emulating the two prefixes w and w' and leading to a vertex t where $\mathcal{P}_1$ has to pick a side corresponding to the two languages $[K_1]$ and $[K_2]$ (Figure 4.4). Now, establishing the $\mathcal{M}$-monotony of $\sqsubseteq$ boils down to invoking an optimal strategy σ in the corresponding game, the crux being that this strategy always picks the same edge in t (i.e., the same side between subarenas corresponding to $[K_1]$ and $[K_2]$) as both prefixes w and w' are deemed equivalent by the memory structure $\mathcal{M}$.

Proof of Proposition 4.5.1. We write $\mathcal{M} = (M, m_{\text{init}}, \alpha_{\text{upd}})$ and we assume that for all finite one-player arenas of $\mathcal{P}_1$, $\mathcal{P}_1$ has an optimal strategy based on $\mathcal{M}$. Let us prove that $\sqsubseteq$ is $\mathcal{M}$-monotone, i.e., that for all $m \in M$, for all regular languages K_1, K_2,

$$\exists w \in L_{m_{\text{init}},m}, [wK_1] \sqsubset [wK_2]$$
$$\implies \forall w' \in L_{m_{\text{init}},m}, [w'K_1] \sqsubseteq [w'K_2]. \quad (4.11)$$

Let $m \in M$ and K_1, K_2 be regular languages. We assume that $K_1, K_2 \neq \emptyset$, otherwise Equation (4.11) holds trivially: if K_1 is empty, the conclusion of the implication is true regardless of K_2; and if K_2 is empty, the premise is false. Now, assume that there exists $w \in L_{m_{\text{init}},m}$ such that $[wK_1] \sqsubset [wK_2]$, and let w' be another finite word in $L_{m_{\text{init}},m}$. We will prove that $[w'K_1] \sqsubseteq [w'K_2]$.

Let $\mathcal{N}^w = (Q^w, C, \Delta^w, q_{\text{init}}^w, F^w)$, $\mathcal{N}^{w'} = (Q^{w'}, C, \Delta^{w'}, q_{\text{init}}^{w'}, F^{w'})$, $\mathcal{N}^{K_1} = (Q^{K_1}, C, \Delta^{K_1}, q_{\text{init}}^{K_1}, F^{K_1})$, and $\mathcal{N}^{K_2} = (Q^{K_2}, C, \Delta^{K_2}, q_{\text{init}}^{K_2}, F^{K_2})$ respectively denote NFAs recognizing languages $\{w\}$, $\{w'\}$, K_1, and K_2. They exist since all these languages are regular. We assume w.l.o.g. that NFA $\mathcal{N}^w$ (resp. $\mathcal{N}^{w'}, \mathcal{N}^{K_1}, \mathcal{N}^{K_2}$) is coaccessible and has only one initial state q_{init}^w (resp. $q_{\text{init}}^{w'}, q_{\text{init}}^{K_1}, q_{\text{init}}^{K_2}$) with no ingoing transition. We can do this since K_1 and K_2 are non-empty. We also assume w.l.o.g. that $\mathcal{N}^w$ (resp. $\mathcal{N}^{w'}$) has only one final state q_{fin}^w (resp. $q_{\text{fin}}^{w'}$) with no outgoing transition. Actually, $\mathcal{N}^w$ and $\mathcal{N}^{w'}$ can be taken as "chains" recognizing a unique word, and being coaccessible and deterministic.

We build an NFA $\mathcal{N} = (Q, C, \Delta, Q_{\text{init}}, F)$ by "merging" states $q_{\text{init}}^{K_1}, q_{\text{init}}^{K_2}, q_{\text{fin}}^w$, and $q_{\text{fin}}^{w'}$. We call this new merged state t. Formally, we build it as follows:

▸ $Q = (Q^w \cup Q^{w'} \cup Q^{K_1} \cup Q^{K_2} \cup \{t\}) \setminus \{q_{\text{init}}^{K_1}, q_{\text{init}}^{K_2}, q_{\text{fin}}^w, q_{\text{fin}}^{w'}\}$;

▶ $Q_{\text{init}} = \{q_{\text{init}}^w, q_{\text{init}}^{w'}\}$ and $F = F^{K_1} \cup F^{K_2}$;

and finally, the transition relation simply takes into account the merging on t:

$$\Delta = \{(q, c, q') \in \Delta^w \cup \Delta^{w'} \cup \Delta^{K_1} \cup \Delta^{K_2} \mid q, q' \notin \{q_{\text{init}}^{K_1}, q_{\text{init}}^{K_2}, q_{\text{fin}}^w, q_{\text{fin}}^{w'}\}\}$$
$$\cup \{(q, c, t) \mid (q, c, q') \in (\Delta^w \cup \Delta^{w'}) \wedge q' \in \{q_{\text{fin}}^w, q_{\text{fin}}^{w'}\}\}$$
$$\cup \{(t, c, q') \mid (q, c, q') \in (\Delta^{K_1} \cup \Delta^{K_2}) \wedge q \in \{q_{\text{init}}^{K_1}, q_{\text{init}}^{K_2}\}\}.$$

This construction is illustrated in Figure 4.4. The language recognized by $\mathcal{N}$ from q_{init}^w is $w(K_1 \cup K_2)$, whereas from $q_{\text{init}}^{w'}$, it is $w'(K_1 \cup K_2)$. Observe that $\mathcal{N}$ is coaccessible since both $\mathcal{N}^{K_1}$ and $\mathcal{N}^{K_2}$ are coaccessible.

Recall that we assume $[wK_1] \sqsubset [wK_2]$. By definition, this implies that $[wK_2] \neq \emptyset$, hence we also have that $[K_2] \neq \emptyset$. From this, we get that t is essential in $\mathcal{N}$ (Lemma 4.2.12). Thus, it is also the case for q_{init}^w and $q_{\text{init}}^{w'}$.

We will now interpret this NFA as a finite arena and use the hypothesis. Let $\mathcal{A} = \text{Arena}(\mathcal{N})$. By Lemma 4.2.12, we have that $\text{col}^\omega(\text{Plays}(\mathcal{A}, q_{\text{init}}^w)) = [w(K_1 \cup K_2)]$ and $\text{col}^\omega(\text{Plays}(\mathcal{A}, q_{\text{init}}^{w'})) = [w'(K_1 \cup K_2)]$. By hypothesis, $\mathcal{P}_1$ has an optimal strategy σ based on $\mathcal{M}$ in $\mathcal{G} = (\mathcal{A}, \sqsubseteq)$. Let $\alpha_{\text{nxt}} \colon V_1 \times M \to E$ be the next-action function of this strategy (where E are the edges of $\mathcal{A}$).

We really need the uniformity of the optimal strategy here, i.e., that the same strategy is as good as possible from multiple initial vertices (here, from q_{init}^w and $q_{\text{init}}^{w'}$). We will come back to the importance of uniformity in Section 4.7.

Let $\rho \in \text{Plays}(\mathcal{A}, q_{\text{init}}^w, \sigma)$ be the only play consistent with strategy σ from q_{init}^w. By definition of $\mathcal{A}$, this play ρ necessarily contains a history $\gamma = e_1 \ldots e_n$ such that $\text{out}(e_n) = t$ and for all $i, 1 \leq i < n, \text{out}(e_i) \neq t$. Observe that $\text{col}^*(\gamma) = w$. Recall that $m = \alpha_{\text{upd}}^*(m_{\text{init}}, w)$ is the memory state reached after reading w since $w \in L_{m_{\text{init}}, m}$. Let $e = \alpha_{\text{nxt}}(t, m)$ be the edge chosen by σ in t when t is visited (t will be visited only once by construction of $\mathcal{A}$).

We will show that e belongs to the part generated by $\mathcal{N}^{K_2}$. By contradiction, assume it belongs to $\mathcal{N}^{K_1}$. Then, $\rho = \gamma \cdot \rho'$, with $\text{col}^\omega(\rho') \in [K_1]$, hence $\text{col}^\omega(\rho) \in [wK_1]$. First, observe that

$$[wK_2] \sqsubseteq \text{col}^\omega(\text{Plays}(\mathcal{A}, q_{\text{init}}^w))$$

since $[wK_2] \subseteq [wK_1] \cup [wK_2], [wK_1] \cup [wK_2] = [w(K_1 \cup K_2)]$ (using Lemma 4.2.11), and, as noted above, $[w(K_1 \cup K_2)] = \text{col}^\omega(\text{Plays}(\mathcal{A}, q_{\text{init}}^w))$. Since σ is optimal from q_{init}^w, we have

$$[wK_2] \sqsubseteq \text{col}^\omega(\rho).$$

One can get from the definition of *optimal* using the UCol-operator that, in this one-player game, σ is as good as possible from q_{init}^w if and only if for all strategies σ' of $\mathcal{P}_1$, $\text{col}^\omega(\text{Plays}(\mathcal{A}, q_{\text{init}}^w, \sigma')) \sqsubseteq \text{col}^\omega(\text{Plays}(\mathcal{A}, q_{\text{init}}^w, \sigma))$.

Finally, as we assumed that $\text{col}^\omega(\rho) \in [wK_1]$, we can conclude that

$$[wK_2] \sqsubseteq [wK_1],$$

which contradicts the hypothesis that $[wK_1] \sqsubset [wK_2]$. Hence, we have established that e belongs to $\mathcal{N}^{K_2}$.

Now let us consider $\rho'' \in \mathsf{Plays}(\mathcal{A}, q_{\text{init}}^{w'}, \sigma)$, the only play consistent with strategy σ from $q_{\text{init}}^{w'}$. Again, by definition of $\mathcal{A}$, this play ρ'' necessarily contains a history $\gamma'' = e_1 \ldots e_n$ such that $\mathsf{out}(e_n) = t$ and for all i, $1 \leq i < n$, $\mathsf{out}(e_i) \neq t$. Observe that $\mathsf{col}^*(\gamma'') = w'$. Since $w' \in L_{m_{\text{init}}, m}$, we also have that $\alpha_{\text{upd}}^*(m_{\text{init}}, w') = m$, i.e., the memory state reached after reading w' is the same as the one reached after reading w. Hence, we have that $e = \alpha_{\text{nxt}}(t, m)$ is exactly the same edge as before, and therefore belongs to $\mathcal{N}^{K_2}$. Thus, $\mathsf{col}^\omega(\rho'') \in [w'K_2]$.

Finally, since σ is also as good as possible from $q_{\text{init}}^{w'}$ and applying the same reasoning as above, we have that

$$[w'K_1] \sqsubseteq [w'K_1] \cup [w'K_2] = [w'(K_1 \cup K_2)]$$
$$= \mathsf{col}^\omega(\mathsf{Plays}(\mathcal{A}, q_{\text{init}}^{w'}))$$
$$\sqsubseteq \mathsf{col}^\omega(\rho'')$$
$$\sqsubseteq [w'K_2],$$

which proves Equation (4.11) and concludes our proof. $\qquad\square$

Selectivity. We now turn to $\mathcal{M}$-selectivity.

> **Proposition 4.5.2** (Necessity of $\mathcal{M}$-selectivity) *Let $\sqsubseteq$ be a preference relation and $\mathcal{M}$ be a memory structure. If for all its finite one-player arenas, for all vertices v of the arena, $\mathcal{P}_1$ has a strategy based on $\mathcal{M}$ that is as good as possible from v, then $\sqsubseteq$ is $\mathcal{M}$-selective.*

The same holds for $\mathcal{P}_2$ and $\sqsubseteq^{-1}$ symmetrically. Observe that uniformity of the strategy is not required here (as opposed to the proof for $\mathcal{M}$-monotone, Proposition 4.5.1).

Our proof bears similarities with the monotone case. We need to establish that Equation (4.8) holds. We first instantiate the four languages involved in it: $\{w\}$, K_1, K_2 and K_3. We take NFAs recognizing them and build an NFA $\mathcal{N}$ that joins them in such a way that, when $\mathcal{N}$ is considered as a finite arena (see Lemma 4.2.12), its plays correspond exactly to the languages of infinite words considered in Equation (4.8). This arena is essentially composed of a chain emulating the prefix w and leading to a vertex t where $\mathcal{P}_1$ can visit sides that generate cycles from K_1 and K_2 — forever or for a finite time — or branch to a side corresponding to K_3 (Figure 4.5). Now, establishing the $\mathcal{M}$-selectivity of $\sqsubseteq$ boils down to invoking an optimal strategy σ in the corresponding game, the crux being that this strategy always picks the same edge in t (i.e., the same side between subarenas corresponding to $[K_1^*]$, $[K_2^*]$ and $[K_3]$) as all cycles on t are deemed equivalent by the memory structure $\mathcal{M}$. The main difference with the previous construction appears in the last sentence: it is now possible to come back to t, possibly infinitely often, and our proof takes that into account.

Proof of Proposition 4.5.2. We write $\mathcal{M} = (M, m_{\text{init}}, \alpha_{\text{upd}})$ and we assume that for all its finite one-player arenas, for all vertices v of the arena, $\mathcal{P}_1$ has a strategy based on $\mathcal{M}$ as good as possible from v. Let us prove that $\sqsubseteq$ is $\mathcal{M}$-selective, i.e., that for all $w \in C^*$, $m = \alpha^*_{\text{upd}}(m_{\text{init}}, w)$, for all regular languages K_1, K_2, K_3 such that $K_1, K_2 \subseteq L_{m,m}$,

$$[w(K_1 \cup K_2)^* K_3] \sqsubseteq [wK_1^*] \cup [wK_2^*] \cup [wK_3]. \tag{4.12}$$

Let $w \in C^*$ and $m = \alpha^*_{\text{upd}}(m_{\text{init}}, w)$. Let K_1, K_2, K_3 be regular languages with $K_1, K_2 \subseteq L_{m,m}$. In the following, we assume all three languages K_1, K_2 and K_3 to be non-empty. Indeed, if K_3 is empty, so is the left-hand side of Equation (4.12), hence it trivially holds. If both K_1 and K_2 are empty, Equation (4.12) compares $[wK_3]$ to itself, hence it trivially holds again. Finally, if K_1 is the only empty language among the three, then Equation (4.12) can be restated as follows:

$$[w(K_1 \cup K_2)^* K_3] = [w(K_2 \cup K_2)^* K_3] \sqsubseteq$$
$$[wK_2^*] \cup [wK_2^*] \cup [wK_3] = [wK_1^*] \cup [wK_2^*] \cup [wK_3],$$

where the middle inequality — the one to prove — involves three non-empty sets. A symmetric argument holds if K_2 is the only empty language. We also assume that K_1 and K_2 do not contain the empty word for technical convenience: this is w.l.o.g. thanks to the Kleene stars used in the regular expressions to consider.

As for monotony, we start by considering NFAs for all these languages: let $\mathcal{N}^w = (Q^w, C, \Delta^w, q^w_{\text{init}}, F^w)$, $\mathcal{N}^{K_1} = (Q^{K_1}, C, \Delta^{K_1}, q^{K_1}_{\text{init}}, F^{K_1})$, $\mathcal{N}^{K_2} = (Q^{K_2}, C, \Delta^{K_2}, q^{K_2}_{\text{init}}, F^{K_2})$, and $\mathcal{N}^{K_3} = (Q^{K_3}, C, \Delta^{K_3}, q^{K_3}_{\text{init}}, F^{K_3})$ respectively denote NFAs recognizing languages $\{w\}$, K_1, K_2 and K_3. They exist since all these languages are regular. We assume w.l.o.g. that NFA $\mathcal{N}^w$ (resp. $\mathcal{N}^{K_1}, \mathcal{N}^{K_2}, \mathcal{N}^{K_3}$) is coaccessible and has only one initial state q^w_{init} (resp. $q^{K_1}_{\text{init}}, q^{K_2}_{\text{init}}, q^{K_3}_{\text{init}}$) with no ingoing transition. We can do this since K_1, K_2, and K_3 are non-empty. We also assume w.l.o.g. that $\mathcal{N}^w$ (resp. $\mathcal{N}^{K_1}, \mathcal{N}^{K_2}$) has only one final state q^w_{fin} (resp. $q^{K_1}_{\text{fin}}, q^{K_2}_{\text{fin}}$) with no outgoing transition. Again $\mathcal{N}^w$ can simply be a "chain" recognizing a unique word, being both coaccessible and deterministic.

Similarly to Proposition 4.5.1, we build an NFA $\mathcal{N} = (Q, C, \Delta, Q_{\text{init}}, F)$ by "merging" states $q^{K_1}_{\text{init}}, q^{K_2}_{\text{init}}, q^{K_3}_{\text{init}}, q^w_{\text{fin}}, q^{K_1}_{\text{fin}}$, and $q^{K_2}_{\text{fin}}$. We call this new merged state t. Formally, we build it as follows:

- $Q = (Q^w \cup Q^{K_1} \cup Q^{K_2} \cup Q^{K_3} \cup \{t\}) \setminus \{q^{K_1}_{\text{init}}, q^{K_2}_{\text{init}}, q^{K_3}_{\text{init}}, q^w_{\text{fin}}, q^{K_1}_{\text{fin}}, q^{K_2}_{\text{fin}}\}$;
- $Q_{\text{init}} = \{q^w_{\text{init}}\}$ and $F = F^{K_3}$;

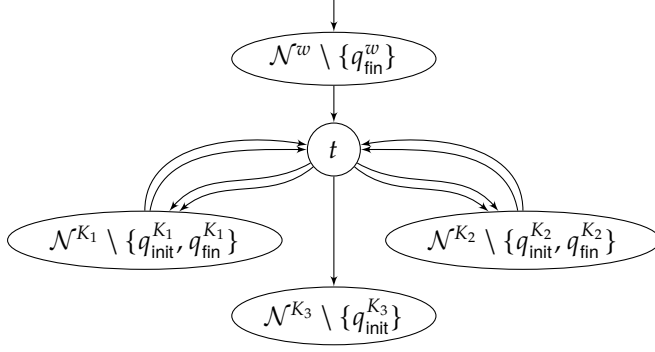and finally, the transition relation simply takes into account the merging

Figure 4.5: NFA $\mathcal{N}$ built to establish $\mathcal{M}$-selectivity.

on t:

$$\Delta = \{(q, c, q') \in \Delta^w \cup \Delta^{K_1} \cup \Delta^{K_2} \cup \Delta^{K_3} \mid$$
$$q, q' \notin \{q_{\text{init}}^{K_1}, q_{\text{init}}^{K_2}, q_{\text{init}}^{K_3}, q_{\text{fin}}^w, q_{\text{fin}}^{K_1}, q_{\text{fin}}^{K_2}\}\}$$
$$\cup \{(q, c, t) \mid (q, c, q') \in (\Delta^w \cup \Delta^{K_1} \cup \Delta^{K_2}) \wedge q' \in \{q_{\text{fin}}^w, q_{\text{fin}}^{K_1}, q_{\text{fin}}^{K_2}\}\}$$
$$\cup \{(t, c, q') \mid (q, c, q') \in (\Delta^{K_1} \cup \Delta^{K_2} \cup \Delta^{K_3}) \wedge q \in \{q_{\text{init}}^{K_1}, q_{\text{init}}^{K_2}, q_{\text{init}}^{K_3}\}\}.$$

This construction is illustrated in Figure 4.5. The language recognized by $\mathcal{N}$ is $w(K_1 \cup K_2)^* K_3$. Observe that $\mathcal{N}$ is coaccessible since $\mathcal{N}^{K_1}$, $\mathcal{N}^{K_2}$ and $\mathcal{N}^{K_3}$ are coaccessible. Also observe that t is essential by construction: by merging the initial and final states of K_1 (resp. K_2), we created cycles on t. Thus, q_{init}^w is also essential.

We will now interpret this NFA as a finite arena and use the hypothesis. Let $\mathcal{A} = \mathsf{Arena}(\mathcal{N})$. By Lemma 4.2.12, we have that $\mathsf{col}^\omega(\mathsf{Plays}(\mathcal{A}, q_{\text{init}}^w)) = [w(K_1 \cup K_2)^* K_3]$. By hypothesis, $\mathcal{P}_1$ has an optimal strategy σ from q_{init}^w, with next-action function $\alpha_{\text{nxt}} \colon V_1 \times M \to E$ (where E are the edges of $\mathcal{A}$).

Let $\rho \in \mathsf{Plays}(\mathcal{A}, q_{\text{init}}^w, \sigma)$ be the only play consistent with σ from q_{init}^w. By optimality from q_{init}^w, we have that

$$[w(K_1 \cup K_2)^* K_3] \sqsubseteq \mathsf{col}^\omega(\rho). \tag{4.13}$$

By definition of $\mathcal{A}$, this play ρ necessarily contains a history $\gamma = e_1 \ldots e_n$ such that $\mathsf{out}(e_n) = t$ and for all i, $1 \le i < n$, $\mathsf{out}(e_i) \ne t$. Observe that $\mathsf{col}^*(\gamma) = w$. Recall that $m = \alpha_{\text{upd}}^*(m_{\text{init}}, w)$ is the memory state reached after reading w since $w \in L_{m_{\text{init}}, m}$. Let $e = \alpha_{\text{nxt}}(t, m)$ be the edge chosen by σ in t when t is first visited. Note that in contrast to the construction in Proposition 4.5.1, t could be visited many times here, and even infinitely often (using cycles from K_1 and K_2). We consider two cases in the following.

First, assume that e belongs to the part of the arena generated by $\mathcal{N}^{K_3}$. Since t (originally $q_{\text{init}}^{K_3}$) has no incoming transition in $\mathcal{N}^{K_3}$, we conclude that ρ never visits t again, and that $\mathsf{col}^\omega(\rho) \in [wK_3]$. By Equation (4.13), we have Equation (4.12).

Now, assume that e belongs to the part of the arena generated by $\mathcal{N}^{K_1}$ (the same reasoning applies symmetrically for $\mathcal{N}^{K_2}$). We want to show that

$\mathsf{col}^\omega(\rho) \in [wK_1^*]$, i.e., that σ never switches to another part of the arena. Two cases are possible: either (a) ρ visits t only once, or (b) ρ visits t at least twice.

Case (a). Since ρ visits t only once and t is the only vertex where the play could switch to a different NFA, we have that $\rho = \gamma \cdot \rho'$ for a suffix ρ' starting in t and entirely contained in $\mathcal{N}^{K_1}$. Hence, we have $\mathsf{col}^\omega(\rho) = w \cdot \mathsf{col}^\omega(\rho')$ with $\mathsf{col}^\omega(\rho') \in [K_1]$. Thus, $\mathsf{col}^\omega(\rho) \in [wK_1] \subseteq [wK_1^*]$.

Case (b). Let $\rho = \gamma \cdot \gamma' \cdot \rho'$, such that γ' ends with the second visit of t. Recall that $w = \mathsf{col}^*(\gamma)$, $m = \alpha_{\mathsf{upd}}^*(m_{\mathsf{init}}, w)$, and $e = \alpha_{\mathsf{nxt}}(t, m)$. Now, by definition of K_1, we have that $\mathsf{col}^*(\gamma') \in L_{m,m}$. Hence, $\alpha_{\mathsf{upd}}^*(m, \mathsf{col}^*(\gamma')) = m$. Intuitively, the memory structure is back to the same memory state after reading the cycle γ'. Therefore, $\sigma(\gamma \cdot \gamma') = \sigma(\gamma) = e$. Iterating this reasoning (as all cycles on t in $\mathcal{N}^{K_1}$ are read as cycles on m in the memory), we conclude that $\rho = \gamma \cdot (\gamma')^\omega$. This implies that $\mathsf{col}^\omega(\rho) \in [wK_1^*]$.

Hence, in both cases, we have that $\mathsf{col}^\omega(\rho) \in [wK_1^*]$. Now, by Equation (4.13), we have Equation (4.12). □

Wrap-up. We have established that the existence of finite-memory optimal strategies based on a memory structure $\mathcal{M}$ in finite one-player games implies both $\mathcal{M}$-monotony and $\mathcal{M}$-selectivity of the preference relation. Next, we consider the converse: we will prove that $\mathcal{M}$-monotony and $\mathcal{M}$-selectivity implies the existence of optimal strategies based on $\mathcal{M}$, not only in one-player games, but even in *two-player* ones, when satisfied by the preference relation *and its inverse*.

4.6 From $\mathcal{M}$-monotony and $\mathcal{M}$-selectivity to strategies based on $\mathcal{M}$

Induction step. To prove the sought implication (Proposition 4.6.3), we first focus on *memoryless* strategies in finite "covered" arenas, as discussed in Section 4.3. Intuitively, a "covered" arena resembles a product arena (with a memory structure): hence studying memoryless strategies on such arenas is very close to studying finite-memory strategies on general arenas.

We will proceed by induction on the number of choices in finite arenas, as sketched in Section 4.3. This induction will require us to mix different Nash equilibria (one for each player) in a proper way to maintain the desired property. For the sake of readability, we thus start by proving the induction step for one player.

For a finite arena $\mathcal{A} = (V, V_1, V_2, E)$, we write $n_\mathcal{A} = |E| - |V|$ for its number of *choices*. We also define the notion of *subarena*: we say that an arena $\mathcal{A}' = (V', V_1', V_2', E')$ is a *subarena* of an arena (V, V_1, V_2, E) if $V_1 = V_1'$, $V_2 = V_2'$, and $E' \subseteq E$. That is, arena $\mathcal{A}'$ is a subarena of $\mathcal{A}$ if it

We recall that as arenas are non-blocking, $|E| \geq |V|$, so $n_\mathcal{A} \geq 0$.

can be obtained from $\mathcal{A}$ by removing some edges of $\mathcal{A}$ (while keeping it non-blocking, as is required to be an arena). We say that a set of arenas $\mathfrak{A}$ is *closed under subarena operation* if for all $\mathcal{A} \in \mathfrak{A}$, for all subarenas $\mathcal{A}'$ of $\mathcal{A}$, $\mathcal{A}' \in \mathfrak{A}$.

> **Lemma 4.6.1** *Let $\sqsubseteq$ be a preference relation, $\mathcal{M}^p$ and $\mathcal{M}^c$ be two memory structures, and $\mathfrak{A}$ be a set of finite arenas closed under subarena operation. We assume that $\sqsubseteq$ is $\mathcal{M}^p$-monotone and $\mathcal{M}^c$-selective, and that for all finite one-player arenas $\mathcal{A} = (V, V_1, V_2, E) \in \mathfrak{A}$ of $\mathcal{P}_2$, for all subsets of vertices $V_{\text{cov}} \subseteq V$ of which $\mathcal{M}^p$ is a prefix-cover and $\mathcal{M}^c$ is a cyclic-cover, $\mathcal{P}_2$ has an optimal strategy from V_{cov}.*
> *Let $n \in \mathbb{N}$. Assume that for all arenas $\mathcal{A}' = (V', V_1', V_2', E') \in \mathfrak{A}$ such that $n_{\mathcal{A}'} < n$, for all subsets of vertices $V'_{\text{cov}} \subseteq V'$ of which $\mathcal{M}^p$ is a prefix-cover and $\mathcal{M}^c$ is a cyclic-cover, there exists a memoryless Nash equilibrium (σ_1', σ_2') from V'_{cov} in $\mathcal{G}' = (\mathcal{A}', \sqsubseteq)$.*
> *Then, for all arenas $\mathcal{A} = (V, V_1, V_2, E) \in \mathfrak{A}$ such that $n_{\mathcal{A}} = n$, for all subsets of vertices $V_{\text{cov}} \subseteq V$ for which $\mathcal{M}^p$ is a prefix-cover and $\mathcal{M}^c$ is a cyclic-cover, there exists a Nash equilibrium (σ_1, σ_2) from V_{cov} in $\mathcal{G} = (\mathcal{A}, W)$ such that σ_1 is memoryless.*

Note that the same holds for $\mathcal{P}_2$ and $\sqsubseteq^{-1}$ symmetrically.

Intuitively, Lemma 4.6.1 states that under the hypotheses of $\mathcal{M}^p$-monotony and $\mathcal{M}^c$-selectivity, if both players can play optimally with memoryless strategies in "small" and "covered" arenas, the same property holds for at least $\mathcal{P}_1$ in "covered" arenas where an additional choice exists. This lemma is focused on proving the existence of an NE in which the strategy of $\mathcal{P}_1$ is memoryless: proving that this holds for both players will be done in Proposition 4.6.3.

As motivated in Section 4.4, we state our result as the existence of memoryless optimal strategies in arenas "covered by $\mathcal{M}$": the existence of strategies based on $\mathcal{M}$ in general arenas will follow (Corollary 4.6.5).

We use two different memory structures, one for monotony (dealing with prefixes) and one for selectivity (dealing with cycles). Naturally, one can use a single combined memory structure using Lemmas 4.3.6 and 4.3.13, but our approach has the advantage of being compositional and highlighting how each property impacts the reasoning in the proof: we will see that they have different uses.

As mentioned above, our proof is essentially an induction step. Starting from a finite arena $\mathcal{A}$ with $n_{\mathcal{A}} = n$ choices, we identify a vertex t in which $\mathcal{P}_1$ has at least two outgoing edges (the proof is symmetric for $\mathcal{P}_2$). By splitting the edges in t into two sets, we obtain two corresponding subarenas $\mathcal{A}_a$ and $\mathcal{A}_b$ such that $n_{\mathcal{A}_a}, n_{\mathcal{A}_b} < n$, along with the corresponding subgames. The induction hypothesis gives us two memoryless Nash equilibria (from V_{cov}) in these subgames: (σ_1^a, σ_2^a) and (σ_1^b, σ_2^b). The arguments can then be

unfolded as follows. First, using $\mathcal{M}^p$-monotony and $\mathcal{M}^p$ being a prefix-cover, we identify one subarena (say $\mathcal{A}_a$) which is clearly at least as good as the other for $\mathcal{P}_1$. Secondly, we build a strategy profile $(\sigma_1^\#, \sigma_2^\#)$, that we claim to be an NE in $\mathcal{G}$, in the following way: $\mathcal{P}_1$ uses strategy σ_1^a (the one from the best subarena) and $\mathcal{P}_2$ reacts to the actions of $\mathcal{P}_1$ by playing the corresponding best-response strategy. I.e., if $\mathcal{P}_1$ plays in $\mathcal{A}_a$, $\mathcal{P}_2$ plays according to σ_2^a, and otherwise $\mathcal{P}_2$ plays according to σ_2^b. Third, it remains to prove the two inequalities of Equation (4.1). The rightmost one is easy, as well as the leftmost one in the subcase where the unique play $\rho \in \mathsf{Plays}(\mathcal{A}, v, \sigma_1^\#, \sigma_2^\#)$ does not visit vertex t: they can both be proved essentially thanks to the induction hypothesis and easy construction arguments. The crux of the proof is thus in the last step: proving that the leftmost inequality holds when the play visits t. This can be achieved thanks to $\mathcal{M}^c$-selectivity and $\mathcal{M}^c$ being a cyclic-cover, Lemma 4.2.11, inherent properties of the preference relation, $\mathcal{A}_a$ being the best subarena thanks to $\mathcal{M}^p$-monotony, and the induction hypothesis.

Proof of Lemma 4.6.1. We write the memory structures from the statement as $\mathcal{M}^p = (M^p, m_{\mathsf{init}}^p, \alpha_{\mathsf{upd}}^p)$ and $\mathcal{M}^c = (M^c, m_{\mathsf{init}}^c, \alpha_{\mathsf{upd}}^c)$.

Let $\mathcal{A} = (V, V_1, V_2, E) \in \mathfrak{A}$ be a finite arena such that $n_\mathcal{A} = n$, and let $V_{\mathsf{cov}} \subseteq V$ be a subset of vertices for which $\mathcal{M}^p$ is a prefix-cover and $\mathcal{M}^c$ is a cyclic-cover. Our goal is to prove that there exists an NE (σ_1, σ_2) from V_{cov} in $(\mathcal{A}, W)$ such that σ_1 is memoryless.

If $\mathcal{A}$ is such that $\mathcal{P}_1$ has no choice (i.e., all vertices in V_1 have only one outgoing edge), then there is a single strategy of $\mathcal{P}_1$ on $\mathcal{A}$, which happens to be memoryless. Arena $\mathcal{A}$ is almost a one-player arena of $\mathcal{P}_2$, as even though $\mathcal{P}_1$ owns some vertices, there is no choice to make in those vertices. We can easily build a bijection between strategies of $\mathcal{P}_2$ in $\mathcal{A}$ and the "true" one-player arena in which ownership of vertices of $\mathcal{P}_1$ is transferred to $\mathcal{P}_2$. By hypothesis, $\mathcal{P}_2$ has a strategy to play in $\mathcal{A}$ that is as good as possible from vertices prefix-covered by $\mathcal{M}^p$ and cyclic-covered by $\mathcal{M}^c$. Thus there is indeed a Nash equilibrium on $\mathcal{A}$ from V_{cov} with the strategy of $\mathcal{P}_1$ being memoryless.

Let us now assume that $\mathcal{P}_1$ has at least one choice and let $t \in V_1$ be a vertex with at least two outgoing edges, i.e., $|\{e \in E \mid \mathsf{in}(e) = t\}| > 1$. We partition $\{e \in E \mid \mathsf{in}(e) = t\}$ in two (non-empty) sets E_a and E_b, and we define two corresponding subarenas, $\mathcal{A}_a = (V, V_1, V_2, E \setminus E_b)$, and $\mathcal{A}_b = (V, V_1, V_2, E \setminus E_a)$, which are in $\mathfrak{A}$ as this set is closed under subarena operation. Observe that it remains true that $\mathcal{M}^p$ is a prefix-cover and $\mathcal{M}^c$ is a cyclic-cover of V_{cov}, both in $\mathcal{A}_a$ and $\mathcal{A}_b$, by definition of prefix-covers and cyclic-covers (intuitively, we quantify universally over fewer histories than in $\mathcal{A}$).

Thus, by induction hypothesis (since $n_{\mathcal{A}_a}, n_{\mathcal{A}_b} < n_\mathcal{A} = n$), we have memoryless NEs from V_{cov} in the subgames $\mathcal{G}_a = (\mathcal{A}_a, \sqsubseteq)$ and $\mathcal{G}_b = (\mathcal{A}_b, \sqsubseteq)$. Let us denote them by (σ_1^j, σ_2^j) for game $\mathcal{G}_j$, $j \in \{a, b\}$.

Since $\mathcal{M}^{\mathsf{p}}$ is a prefix-cover of V_{cov} in $\mathcal{A}$, there exists $m_t^{\mathsf{p}} \in M^{\mathsf{p}}$ such that, for all $\gamma \in \mathsf{Hists}(\mathcal{A})$ such that $\mathsf{in}(\gamma) \in V_{\mathsf{cov}}$, $\mathsf{out}(\gamma) = t$, and such that for all γ' proper prefix of γ, $\mathsf{out}(\gamma') \neq t$, we have $\alpha_{\mathsf{upd}}^*(m_{\mathsf{init}}^{\mathsf{p}}, \mathsf{col}^*(\gamma)) = m_t^{\mathsf{p}}$. Now, let $K_j^{\mathsf{p}} = \mathsf{col}^\omega\big(\mathsf{Hists}(\mathcal{A}_j, t, \sigma_2^j)\big)$, for $j \in \{a, b\}$, that is, K_j^{p} contains all (projections to colors of) histories consistent with σ_2^j and starting in t in subarena $\mathcal{A}_j$.

By $\mathcal{M}^{\mathsf{p}}$-monotony, we can deduce that we have

$$\forall w \in L_{m_{\mathsf{init}}^{\mathsf{p}}, m_t^{\mathsf{p}}}^{\mathsf{p}}, \, [wK_a^{\mathsf{p}}] \sqsubseteq [wK_b^{\mathsf{p}}],$$

$$\text{or } \forall w \in L_{m_{\mathsf{init}}^{\mathsf{p}}, m_t^{\mathsf{p}}}^{\mathsf{p}}, \, [wK_b^{\mathsf{p}}] \sqsubseteq [wK_a^{\mathsf{p}}] \tag{4.14}$$

where $L_{m_{\mathsf{init}}^{\mathsf{p}}, m_t^{\mathsf{p}}}^{\mathsf{p}}$ stands for the usual language of sequences of colors read from $m_{\mathsf{init}}^{\mathsf{p}}$ to m_t^{p}, the additional superscript being used to highlight that we are considering memory structure $\mathcal{M}^{\mathsf{p}}$ here. From now on, we assume w.l.o.g. that Equation (4.14) holds, i.e., that $\forall w \in L_{m_{\mathsf{init}}^{\mathsf{p}}, m_t^{\mathsf{p}}}^{\mathsf{p}}, \, [wK_b^{\mathsf{p}}] \sqsubseteq [wK_a^{\mathsf{p}}]$. Intuitively, this means that, for $\mathcal{P}_1$, committing to the subarena $\mathcal{A}_a$ is always at least as good as committing to the subarena $\mathcal{A}_b$. Note that this does not imply anything with regard to alternating between the two subarenas, which could for now be beneficial: we will deal with that soon thanks to $\mathcal{M}^{\mathsf{c}}$-selectivity.

Let us define the strategy $\sigma_1^\#$ of $\mathcal{P}_1$ on arena $\mathcal{A}$ as $\sigma_1^\# = \sigma_1^a$, the strategy used in the NE from V_{cov} in the subgame $\mathcal{G}_a$ (we chose this one because of assumption (4.14): $\mathcal{G}_a$ is the better subgame of the two for $\mathcal{P}_1$). Strategy $\sigma_1^\#$ is thus memoryless by definition. Note that $\sigma_1^\#$ is well-defined on $\mathcal{A}$ even though the original strategy was on $\mathcal{A}_a$, since it is memoryless (i.e., whether the prefix did visit $\mathcal{A}_b$ or not does not matter). Now, we define a corresponding strategy $\sigma_2^\#$ for $\mathcal{P}_2$ in $\mathcal{G}$ that uses a small amount of (chaotic) memory, as follows: for all $\gamma \in \mathsf{Hists}_2(\mathcal{A})$,

$$\sigma_2^\#(\gamma) = \begin{cases} \sigma_2^a(\gamma) & \text{if } \gamma \text{ never visited } t, \\ \sigma_2^a(\gamma) & \text{if the last visit of } t \text{ was followed by an edge in } \mathcal{A}_a, \\ \sigma_2^b(\gamma) & \text{otherwise.} \end{cases}$$

Again this strategy is well-defined on $\mathcal{A}$. Our goal is to show that the strategy profile $(\sigma_1^\#, \sigma_2^\#)$ is an NE from all vertices in V_{cov} in the larger game $\mathcal{G}$. In particular, Lemma 4.2.6 implies that this pair of strategies are optimal from V_{cov}.

Formally, we will establish that for all $v \in V_{\mathsf{cov}}$, for all strategies σ_1 of $\mathcal{P}_1$ and strategies σ_2 of $\mathcal{P}_2$ on $\mathcal{A}$, we have

$$\mathsf{col}^\omega(\mathsf{Plays}(\mathcal{A}, v, \sigma_1, \sigma_2^\#)) \sqsubseteq \mathsf{col}^\omega(\mathsf{Plays}(\mathcal{A}, v, \sigma_1^\#, \sigma_2^\#))$$

$$\sqsubseteq \mathsf{col}^\omega(\mathsf{Plays}(\mathcal{A}, v, \sigma_1^\#, \sigma_2)). \tag{4.15}$$

We begin with the rightmost inequality of Equation (4.15). Let $v \in V_{\text{cov}}$ and let σ_2 be an arbitrary strategy for $\mathcal{P}_2$ on $\mathcal{A}$. We denote by $\sigma_2[\mathcal{A}_a]$ its restriction to (histories of) $\mathcal{A}_a$: note that this strategy is well-defined as only edges belonging to $\mathcal{P}_1$ have been removed in $\mathcal{A}_a$.

We have

$$\text{col}^{\omega}(\text{Plays}(\mathcal{A}, v, \sigma_1^{\#}, \sigma_2^{\#}))$$
$$= \text{col}^{\omega}(\text{Plays}(\mathcal{A}_a, v, \sigma_1^a, \sigma_2^a)) \qquad \text{as these strategies stay in } \mathcal{A}_a$$
$$\sqsubseteq \text{col}^{\omega}(\text{Plays}(\mathcal{A}_a, v, \sigma_1^a, \sigma_2[\mathcal{A}_a])) \quad \text{as } (\sigma_1^a, \sigma_2^a) \text{ is an NE from } v \text{ in } \mathcal{A}_a$$
$$= \text{col}^{\omega}(\text{Plays}(\mathcal{A}, v, \sigma_1^{\#}, \sigma_2)) \qquad \text{as these strategies stay in } \mathcal{A}_a,$$

hence the rightmost inequality holds.

Now, consider the leftmost inequality of Equation (4.15). Let $v \in V_{\text{cov}}$ and let σ_1 be an arbitrary strategy of $\mathcal{P}_1$ on $\mathcal{A}$. Let $\rho \in \text{Plays}(\mathcal{A}, v, \sigma_1, \sigma_2^{\#})$ be the only play consistent with σ_1 and $\sigma_2^{\#}$ from v. We first consider the case where ρ never visits t. If this is the case, then ρ is also a play of $\mathcal{A}_a$. Let σ_1' be a strategy of $\mathcal{P}_1$ on $\mathcal{A}_a$ that mimics σ_1 on all histories that belong to $\mathcal{A}_a$, except the ones ending in t, where it plays an arbitrary edge in E_a. We have

$$\text{col}^{\omega}(\rho)$$
$$= \text{col}^{\omega}(\text{Plays}(\mathcal{A}, v, \sigma_1, \sigma_2^{\#}))$$
$$= \text{col}^{\omega}(\text{Plays}(\mathcal{A}_a, v, \sigma_1', \sigma_2^a)) \quad \text{because } \rho \text{ stays in } \mathcal{A}_a \text{ and never visits } t,$$
$$\sqsubseteq \text{col}^{\omega}(\text{Plays}(\mathcal{A}_a, v, \sigma_1^a, \sigma_2^a)) \quad \text{because } (\sigma_1^a, \sigma_2^a) \text{ is an NE from } v \text{ in } \mathcal{A}_a,$$
$$= \text{col}^{\omega}(\text{Plays}(\mathcal{A}, v, \sigma_1^{\#}, \sigma_2^{\#})) \qquad \text{because these strategies stay in } \mathcal{A}_a,$$

hence the leftmost inequality holds in the case where ρ never visits t.

It remains to consider the case where ρ does visit t. Observe that for the moment, we have not used $\mathcal{M}^c$-selectivity and the fact that $v \in V_{\text{cov}}$: they will be crucial to solve this (more complex) case.

We define $K_j^c = \text{col}^*(\{\gamma \in \text{Hists}(\mathcal{A}_j, t, \sigma_2^j) \mid \text{out}(\gamma) = t\})$, for $j \in \{a, b\}$, that is, K_j^c contains all (projections to colors of) cycles on t consistent with σ_2^j (i.e., the strategy from the subgame NE) in subarena $\mathcal{A}_j$. Since the unique play $\rho \in \text{Plays}(\mathcal{A}, v, \sigma_1, \sigma_2^{\#})$ visits t at least once, we write $\rho = \gamma \cdot \rho'$ for γ, the prefix ending with the first visit of t. Let $w = \text{col}^*(\gamma)$. Observe that

$$\text{col}^{\omega}(\rho) \in [w(K_a^c \cup K_b^c)^*(K_a^p \cup K_b^p)]$$

since $\sigma_2^{\#}$ alternates between σ_2^a and σ_2^b depending on what $\mathcal{P}_1$ plays in t. Intuitively, either ρ cycles infinitely often on t using cycles of $(K_a^c \cup K_b^c)$, or it does it for a while, then switches to $(K_a^p \cup K_b^p)$, which induces that ρ commits to a subarena. Thus, we trivially have $\text{col}^{\omega}(\rho) \sqsubseteq [w(K_a^c \cup K_b^c)^*(K_a^p \cup K_b^p)]$.

Since $\mathcal{M}^c$ is a cyclic-cover of V_{cov} in $\mathcal{A}$, and γ starts in V_{cov}, we know that for $m^c = \widehat{\alpha_{\text{upd}}^c}(m_{\text{init}}^c, \text{col}^*(\gamma))$, and for all $\gamma' \in \text{Hists}(\mathcal{A})$ such that

$in(\gamma') = out(\gamma') = t$, we have $\widehat{\alpha^{c}_{upd}}(m^{c}, col^{*}(\gamma')) = m^{c}$. That is, all cycles on t are read as cycles on m^{c} in $\mathcal{M}^{c}$. This implies that $K^{c}_{a}, K^{c}_{b} \subseteq L_{m^{c}, m^{c}}$. Knowing that, we can invoke the $\mathcal{M}^{c}$-selectivity (Equation (4.8)) of the preference relation to obtain

$$col^{\omega}(\rho) \sqsubseteq [w(K^{c}_{a})^{*}] \cup [w(K^{c}_{b})^{*}] \cup [w(K^{p}_{a} \cup K^{p}_{b})].$$

Using Lemma 4.2.11, we have

$$col^{\omega}(\rho) \sqsubseteq [w(K^{c}_{a})^{*}] \cup [w(K^{c}_{b})^{*}] \cup [wK^{p}_{a}] \cup [wK^{p}_{b}].$$

Observe that $[w(K^{c}_{j})^{*}] \subseteq [wK^{p}_{j}]$, for $j \in \{a, b\}$. Hence, we have

$$col^{\omega}(\rho) \sqsubseteq [wK^{p}_{a}] \cup [wK^{p}_{b}].$$

Now, recall that using the $\mathcal{M}^{p}$-monotony of $\sqsubseteq$, we assumed that $\forall w \in L^{p}_{m^{p}_{init}, m^{p}_{t}}, [wK^{p}_{b}] \sqsubseteq [wK^{p}_{a}]$. Since γ starts in V_{cov}, ends in its first visit to t and $\mathcal{M}^{p}$ is a prefix-cover of V_{cov}, this inequality is in particular true for $w = col^{*}(\gamma)$. Hence, we have

$$col^{\omega}(\rho) \sqsubseteq [wK^{p}_{a}].$$

Now, recall that $(\sigma^{a}_{1}, \sigma^{a}_{2})$ is an NE from V_{cov} in $\mathcal{G}_{a}$. Recall also that w represents the history up to the first visit of t consistent with $(\sigma_{1}, \sigma^{\#}_{2})$; it is also consistent with $(\sigma_{1}, \sigma^{a}_{2})$ since $\sigma^{\#}_{2}$ follows σ^{a}_{2} up to the first visit of t. Hence, we also have

$$[wK^{p}_{a}] \subseteq col^{\omega}(\mathsf{Plays}(\mathcal{A}_{a}, v, \sigma^{a}_{2})) \sqsubseteq col^{\omega}(\mathsf{Plays}(\mathcal{A}_{a}, v, \sigma^{a}_{1}, \sigma^{a}_{2})).$$

Therefore,

$$col^{\omega}(\rho) \sqsubseteq col^{\omega}(\mathsf{Plays}(\mathcal{A}_{a}, v, \sigma^{a}_{1}, \sigma^{a}_{2}))$$
$$= col^{\omega}(\mathsf{Plays}(\mathcal{A}, v, \sigma^{\#}_{1}, \sigma^{\#}_{2})) \quad \text{because these strategies stay in } \mathcal{A}_{a}.$$

Recalling that ρ is the only play in $\mathsf{Plays}(\mathcal{A}, v, \sigma_{1}, \sigma^{\#}_{2})$, we are done with proving the leftmost inequality of Equation (4.15).

Summing up our arguments, we have established that the couple of strategies $(\sigma^{\#}_{1}, \sigma^{\#}_{2})$ is indeed a Nash equilibrium in $\mathcal{G}$ from V_{cov}. Note that this in particular implies, via Lemma 4.2.6, that $\sigma^{\#}_{1}$ is a memoryless optimal strategy in $\mathcal{G}$ from V_{cov}. □

Memoryless Nash equilibria.   We are now armed to establish the implication sketched earlier. As motivated before, we first state the result in the context of memoryless NEs on finite "covered" arenas; the finite-memory case on general finite arenas will follow straightforwardly. We first show the result for finite one-player arenas, and use it to obtain the two-player case.

Lemma 4.6.2 *Let $\sqsubseteq$ be a preference relation and $\mathcal{M}^p$, $\mathcal{M}^c$ be two memory structures. Assume that $\sqsubseteq$ is $\mathcal{M}^p$-monotone and $\mathcal{M}^c$-selective. Then, for all finite one-player arenas $\mathcal{A} = (V, V_1, V_2, E)$ of $\mathcal{P}_1$, for all subsets of vertices $V_{cov} \subseteq V$ for which $\mathcal{M}^p$ is a prefix-cover and $\mathcal{M}^c$ is a cyclic-cover, there exists a memoryless optimal strategy σ_1 from V_{cov} in $\mathcal{G} = (\mathcal{A}, \sqsubseteq)$.*

Proof. Let $\mathfrak{A}$ be the set of all finite one-player arenas of $\mathcal{P}_1$, which is closed under subarena operation.

We proceed by induction on the number of choices in the arena. The base case, $n_{\mathcal{A}} = 0$, is trivial. Let $n \in \mathbb{N} \setminus \{0\}$ and assume the result holds for $n_{\mathcal{A}} < n$. Let $\mathcal{A} = (V, V_1, V_2, E) \in \mathfrak{A}$ be a one-player arena of $\mathcal{P}_1$ such that $n_{\mathcal{A}} = n$, and let $V_{cov} \subseteq V$ be a subset of vertices for which $\mathcal{M}^p$ is a prefix-cover and $\mathcal{M}^c$ is a cyclic-cover. We can invoke Lemma 4.6.1 (note that as we only consider the one-player arenas of $\mathcal{P}_1$, the existence of a Nash equilibrium coincides with the existence of an optimal strategy for $\mathcal{P}_1$ — see Remark 4.2.8), and obtain an optimal strategy for $\mathcal{P}_1$ from V_{cov}. $\qquad\square$

We are now ready for the two-player case, which requires monotony and selectivity assumptions for both relation $\sqsubseteq$ and relation $\sqsubseteq^{-1}$.

Proposition 4.6.3 *Let $\sqsubseteq$ be a preference relation and $\mathcal{M}_1^p$, $\mathcal{M}_2^p$, $\mathcal{M}_1^c$ and $\mathcal{M}_2^c$ be four memory structures. Assume that $\sqsubseteq$ is $\mathcal{M}_1^p$-monotone and $\mathcal{M}_1^c$-selective, and that $\sqsubseteq^{-1}$ is $\mathcal{M}_2^p$-monotone and $\mathcal{M}_2^c$-selective. Then, for all finite arenas $\mathcal{A} = (V, V_1, V_2, E)$, for all subsets of vertices $V_{cov} \subseteq V$ for which $\mathcal{M}_1^p$ and $\mathcal{M}_2^p$ are prefix-covers, and $\mathcal{M}_1^c$ and $\mathcal{M}_2^c$ are cyclic-covers, there exists a memoryless Nash equilibrium (σ_1, σ_2) from V_{cov} in $\mathcal{G} = (\mathcal{A}, \sqsubseteq)$.*

We want to keep our results as general and compositional as possible; hence, we consider different memory structures for the two players. As argued before, one can always take a single structure for the two players, as well as for the two notions, by taking their product and using Lemmas 4.3.6 and 4.3.13.

As discussed previously, this theorem in particular implies the existence of memoryless optimal strategies from V_{cov} for both players (via Lemma 4.2.6).

It is fairly straightforward to prove Proposition 4.6.3 once Lemma 4.6.1 is established: the main idea is to invoke Lemma 4.6.1 for both players while doing the induction, and obtain two Nash equilibria, both of which being memoryless for only one player. Then, to conclude, we resort to Lemma 4.2.4 which gives us the possibility to mix these two NEs into one that is now memoryless for *both* players.

Remark 4.6.4 Recall that a crucial hypothesis for Lemma 4.2.4 to hold is that our games are *zero-sum*, i.e., that we consider $\sqsubseteq$ and its inverse

relation $\sqsubseteq^{-1}$. It is quite interesting to observe that our use of Lemma 4.2.4 is the only circumstance in which this hypothesis matters (and it is indeed essential) in all our reasoning. In other words, most of our arguments would hold for two different preference relations $\sqsubseteq_1$ and $\sqsubseteq_2$ (for respectively $\mathcal{P}_1$ and $\mathcal{P}_2$), without the hypothesis that $\sqsubseteq_2$ equals $(\sqsubseteq_1)^{-1}$. The problem would be that we cannot mix the two equilibria in a single equilibrium with both strategies being memoryless — while we do need it in the hypothesis of the induction step in Lemma 4.6.1. Whether the same reasoning can be extended to (general) Nash equilibria by adapting Lemma 4.6.1 to take into account the unavoidable blow-up of memory is a question we leave open for future work. Note that the memory bounds would be huge in any case: as the induction unrolls, the memory needed in the equilibria would build up (essentially one bit of memory is added at each call of the induction step in our easier setting, which is then discarded thanks to Lemma 4.6.1).

To be more precise: we wrote everything in the antagonistic setting, but Equation (4.1) can be written as two inequalities in the general setting — $\mathrm{col}^\omega(\mathrm{Plays}(\mathcal{A}, v, \sigma'_1, \sigma_2)) \sqsubseteq_1 \mathrm{col}^\omega(\mathrm{Plays}(\mathcal{A}, v, \sigma_1, \sigma_2))$ and $\mathrm{col}^\omega(\mathrm{Plays}(\mathcal{A}, v, \sigma_1, \sigma'_2)) \sqsubseteq_2 \mathrm{col}^\omega(\mathrm{Plays}(\mathcal{A}, v, \sigma_1, \sigma_2))$ — and all our previous reasoning can be rewritten accordingly.

Proof of Proposition 4.6.3. We consider the set $\mathfrak{A}$ of all finite arenas, which is closed under subarena operation. By Lemma 4.6.2, we immediately obtain that for all finite one-player arenas $\mathcal{A} = (V, V_1, V_2, E)$ of $\mathcal{P}_1$ (resp. of $\mathcal{P}_2$), for all subsets of vertices $V_{\mathrm{cov}} \subseteq V$ for which $\mathcal{M}_1^{\mathrm{p}}$ and $\mathcal{M}_2^{\mathrm{p}}$ are prefix-covers, and $\mathcal{M}_1^{\mathrm{c}}$ and $\mathcal{M}_2^{\mathrm{c}}$ are cyclic-covers, $\mathcal{P}_1$ (resp. $\mathcal{P}_2$) has an optimal strategy from V_{cov}.

We will proceed by induction on the number of choices in the arena, as described before. The base case, $n_{\mathcal{A}} = 0$, is trivial. Now let $n \in \mathbb{N} \setminus \{0\}$ and assume the result holds for $n_{\mathcal{A}} < n$. Let $\mathcal{A} = (V, V_1, V_2, E) \in \mathfrak{A}$ be an arena such that $n_{\mathcal{A}} = n$, and let $V_{\mathrm{cov}} \subseteq V$ be a subset of vertices for which $\mathcal{M}_1^{\mathrm{p}}$ and $\mathcal{M}_2^{\mathrm{p}}$ are prefix-covers, and $\mathcal{M}_1^{\mathrm{c}}$ and $\mathcal{M}_2^{\mathrm{c}}$ are cyclic-covers.

Focusing on $\mathcal{P}_1$ and $\sqsubseteq$, we invoke Lemma 4.6.1 (using $\mathcal{M}_1^{\mathrm{p}} \otimes \mathcal{M}_2^{\mathrm{p}}$ and $\mathcal{M}_1^{\mathrm{c}} \otimes \mathcal{M}_2^{\mathrm{c}}$, and the induction hypothesis) and obtain an NE $(\sigma_1^\spadesuit, \sigma_2^\spadesuit)$ from V_{cov} in $\mathcal{G} = (\mathcal{A}, \sqsubseteq)$ such that $\sigma_1^\spadesuit$ is memoryless. Symmetrically, focusing on $\mathcal{P}_2$ and $\sqsubseteq^{-1}$, we invoke Lemma 4.6.1 (using $\mathcal{M}_1^{\mathrm{p}} \otimes \mathcal{M}_2^{\mathrm{p}}$ and $\mathcal{M}_1^{\mathrm{c}} \otimes \mathcal{M}_2^{\mathrm{c}}$, and the induction hypothesis) and obtain an NE $(\sigma_1^\clubsuit, \sigma_2^\clubsuit)$ from V_{cov} in $\mathcal{G} = (\mathcal{A}, \sqsubseteq)$ such that $\sigma_2^\clubsuit$ is memoryless.

To conclude, it suffices to use Lemma 4.2.4: $(\sigma_1^\spadesuit, \sigma_2^\spadesuit)$ can be mixed with $(\sigma_1^\clubsuit, \sigma_2^\clubsuit)$ into an NE $(\sigma_1^\spadesuit, \sigma_2^\clubsuit)$, which is now memoryless for *both* players. This concludes our induction step and our proof. □

Finite-memory Nash equilibria based on $\mathcal{M}$. Finally, we conclude this section by establishing our result as a corollary.

Corollary 4.6.5 *Let $\sqsubseteq$ be a preference relation and $\mathcal{M}_1^{\mathrm{p}}, \mathcal{M}_2^{\mathrm{p}}, \mathcal{M}_1^{\mathrm{c}}$ and $\mathcal{M}_2^{\mathrm{c}}$ be four memory structures. Assume that $\sqsubseteq$ is $\mathcal{M}_1^{\mathrm{p}}$-monotone and $\mathcal{M}_1^{\mathrm{c}}$-selective and that $\sqsubseteq^{-1}$ is $\mathcal{M}_2^{\mathrm{p}}$-monotone and $\mathcal{M}_2^{\mathrm{c}}$-selective. Then, for all finite arenas $\mathcal{A}$, there exists a Nash equilibrium (σ_1, σ_2) based on memory structure $\mathcal{M} =$*

$$\mathcal{M}_1^{\mathsf{p}} \otimes \mathcal{M}_2^{\mathsf{p}} \otimes \mathcal{M}_1^{\mathsf{c}} \otimes \mathcal{M}_2^{\mathsf{c}} \text{ in } \mathcal{G} = (\mathcal{A}, \sqsubseteq).$$

The proof of this corollary follows directly from the previous results. We build the joint memory structure $\mathcal{M}$ as defined above. By Lemmas 4.3.6 and 4.3.13, we can invoke Proposition 4.6.3 on the product arena $\mathcal{A} \ltimes \mathcal{M}$ and obtain a memoryless NE on it, or equivalently, an NE based on $\mathcal{M}$ on the original arena, through Lemma 4.2.9.

Proof of Corollary 4.6.5. Let $\mathcal{A} = (V, V_1, V_2, E)$ be a finite arena. We define $\mathcal{M} = (M, m_{\mathsf{init}}, \alpha_{\mathsf{upd}}) = \mathcal{M}_1^{\mathsf{p}} \otimes \mathcal{M}_2^{\mathsf{p}} \otimes \mathcal{M}_1^{\mathsf{c}} \otimes \mathcal{M}_2^{\mathsf{c}}$, the joint memory structure. By Lemma 4.3.6, $\sqsubseteq$ and $\sqsubseteq^{-1}$ are both $\mathcal{M}$-monotone and $\mathcal{M}$-selective.

Consider the product arena $\mathcal{A}' = \mathcal{A} \ltimes \mathcal{M}$, as defined in Subsection 4.2.3. Recall that $V' = V \times M$. By Lemma 4.3.13, the set of vertices $V'_{\mathsf{cov}} = V \times \{m_{\mathsf{init}}\} \subseteq V'$ is both prefix-covered and cyclic-covered by $\mathcal{M}$.

Putting the last two arguments together, we invoke Proposition 4.6.3 on $\mathcal{A}'$ and obtain a memoryless Nash equilibrium (σ_1', σ_2') from V'_{cov} in $\mathcal{G}' = (\mathcal{A}', \sqsubseteq)$.

To conclude, it suffices to use Lemma 4.2.9 (stated using NE, as discussed in Remark 4.2.10): the memoryless NE (σ_1', σ_2') in the product game $\mathcal{G}'$ can be seen as an NE (σ_1, σ_2) based on $\mathcal{M}$ in $\mathcal{G} = (\mathcal{A}, \sqsubseteq)$. $\square$

We can also formulate a version of this last result focusing on one-player arenas.

> **Corollary 4.6.6** *Let $\sqsubseteq$ be a preference relation and $\mathcal{M}^{\mathsf{p}}$, $\mathcal{M}^{\mathsf{c}}$ be two memory structures. Assume that $\sqsubseteq$ is $\mathcal{M}^{\mathsf{p}}$-monotone and $\mathcal{M}^{\mathsf{c}}$-selective. Then, for all finite one-player arenas $\mathcal{A} = (V, V_1, V_2, E)$ of $\mathcal{P}_1$, there exists an optimal strategy σ_1 in $\mathcal{G} = (\mathcal{A}, \sqsubseteq)$, based on $\mathcal{M} = \mathcal{M}^{\mathsf{p}} \otimes \mathcal{M}^{\mathsf{c}}$.*

Sketch of proof. The proof is very similar to (and easier than) the proof of Corollary 4.6.5, but uses the one-player implication of Lemma 4.6.2 instead of Proposition 4.6.3. $\square$

4.7 Digression: the cost of uniformity

In Chapter 2, we defined an optimal strategy for a qualitative objective as one that wins *from every vertex in the winning region*, and we have generalized it for preference relations to a strategy *that plays as well as possible from every vertex*. This requirement is sometimes called *uniformity*.

We give a few general situations showing the impact of uniformity on the complexity of strategies, the gist of this section being that uniformity is *almost for free* in two-player games on graphs.
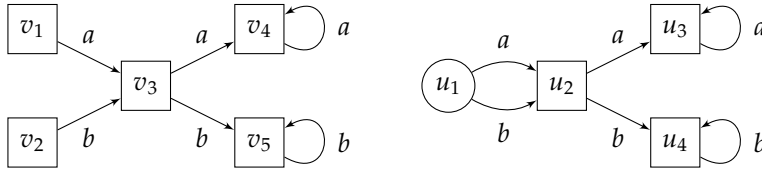
One-player arenas. We first focus our attention on the cost of uniformity in one-player arenas. In one-player arenas, requiring uniformity may demand more complex strategies in general. We have shown that the conjunction of $\mathcal{M}$-monotony and $\mathcal{M}$-selectivity of a preference relation $\sqsubseteq$ was equivalent to the existence of optimal strategies based on $\mathcal{M}$ for $\mathcal{P}_1$ in its finite one-player arenas (Theorem 4.4.3). As it turns out, one can go further down: in [BFMM11, Theorem 4], it is shown that an objective W is $\mathcal{M}_{\mathsf{triv}}$-selective if and only if for all finite one-player arenas of $\mathcal{P}_1$, from all vertices of the winning region, $\mathcal{P}_1$ has a memoryless winning strategy. In other words, $\mathcal{M}_{\mathsf{triv}}$-selectivity is equivalent to the existence of (non-necessarily uniform!) memoryless winning strategies in finite one-player arenas for $\mathcal{P}_1$. This means that $\mathcal{M}_{\mathsf{triv}}$-selectivity guarantees the existence of memoryless winning strategies in one-player arenas, and that $\mathcal{M}_{\mathsf{triv}}$-monotony is then needed to "uniformize" them while keeping them memoryless.

We give a concrete example: we studied objective $W = \mathsf{Reach}(a) \cap \mathsf{Reach}(b)$ in Subsection 4.4.2, with $C = \{a, b\}$. We have seen that its complement $\overline{W} = \mathsf{Safe}(a) \cup \mathsf{Safe}(b)$ is $\mathcal{M}_{\mathsf{triv}}$-selective. With [BFMM11, Theorem 4], this means that $\mathcal{P}_2$ has a memoryless winning strategy from every vertex of its winning region in its finite one-player arenas. However, $\overline{W}$ is not $\mathcal{M}_{\mathsf{triv}}$-monotone: by Theorem 4.4.3, this means that $\mathcal{P}_2$ does not have memoryless optimal (i.e., uniformly winning) strategies in its one-player arenas. We can illustrate it on a small arena: we consider the one-player arena from Figure 4.6 (left). All strategies are winning for $\mathcal{P}_2$ from v_3, v_4, and v_5. From v_1, $\mathcal{P}_2$ wins by going to v_4 and inducing word a^ω; from v_2, $\mathcal{P}_2$ needs to go to v_5 to induce word b^ω. From each fixed vertex, $\mathcal{P}_2$ wins with a memoryless strategy. Yet, $\mathcal{P}_2$ needs memory to win uniformly from v_1 and v_2, as the optimal choice in v_3 depends on what was seen previously.

In two-player arenas, $\mathcal{P}_2$ needs memory to win, even from a fixed vertex; $\mathcal{P}_1$ can emulate the non-uniformity from v_1 and v_2 in a single vertex u_1 in Figure 4.6 (right).

Two-player arenas. We would now like to investigate the impact from requiring uniformity on memory requirements in two-player arenas, and more specifically on arena-independent finite-memory determinacy. For simplicity, we focus on qualitative objectives $W \subseteq C^\omega$.

[BFMM11]: Bianco et al. (2011), *Exploring the boundary of half-positionality*

We recall that a strategy based on $\mathcal{M}_{\mathsf{triv}}$ is exactly a memoryless strategy.

The proof of [BFMM11, Theorem 4] is also by induction on the number of edges, and it could be generalized to deal with arbitrary memory structures $\mathcal{M}$ using an induction on covered arenas.
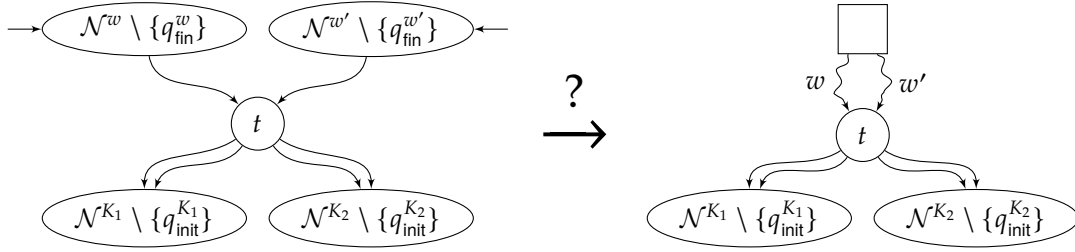
Figure 4.7: Alternative proof attempt for Proposition 4.5.1: replace uniform strategies in one-player arenas by non-uniform strategies in two-player arenas?

In this chapter, there is one proof where we used as an assumption the uniformity of the strategies: the one of Proposition 4.5.1. In this proof, we used the existence of optimal strategies in the class of finite one-player arenas of the kind depicted in Figure 4.7 (left). Could this proof work by replacing the existence of uniform strategies in one-player arenas with *non-uniform strategies*? This is not possible if we keep the strong "one-player arenas" requirement (see counterexample $\mathsf{Safe}(a) \cup \mathsf{Safe}(b)$ above), but is it possible with a two-player arena? Through this other route, we would not get the one-to-two-player lift, but we would get uniformity of strategies for free.

This is tempting; can we just give *the opponent* the choice of the initial word, keeping the same construction elsewhere? This works in almost every situation; an issue only arises when w or w' is the empty word ε! In the original proof using one-player arenas, this would mean that t is taken as one of the two interesting initial vertices, which works. However, there is no way to replicate that from just one initial vertex with the construction on the right of Figure 4.7. If we can replace the empty word with an "equivalent" non-empty word w_ε (in the sense that $\varepsilon \sim w_\varepsilon$; ε and w_ε have the same winning continuations), then the proof carries out; if not, it may not work, as we will illustrate in the upcoming Example 4.7.2. We first state the general result.

Lemma 4.7.1 *Let $W \subseteq C^\omega$ be an objective such that there exists $w_\varepsilon \in C^+$ with $\varepsilon \sim w_\varepsilon$. Let $\mathcal{M}$ be a memory structure. Both players have a (non-necessarily uniform) winning strategy based on $\mathcal{M}$ from every vertex of their winning regions in finite two-player arenas if and only if both players have (uniformly) optimal strategies based on $\mathcal{M}$ in finite two-player arenas.*

Proof. The implication from right to left is trivial. For the implication from left to right, we go through $\mathcal{M}$-monotony and $\mathcal{M}$-selectivity. If both players have a (non-uniform) winning strategy based on $\mathcal{M}$ from every vertex of their winning regions in finite two-player arenas, then W and $\overline{W}$ are $\mathcal{M}$-monotone (same proof as Proposition 4.5.1, replacing the construction with the one on the right of Figure 4.7 and replacing w or w' by w_ε if needed) and $\mathcal{M}$-selective (Proposition 4.5.2). Then, from $\mathcal{M}$-monotony and $\mathcal{M}$-selectivity of W and $\overline{W}$, we obtain the existence
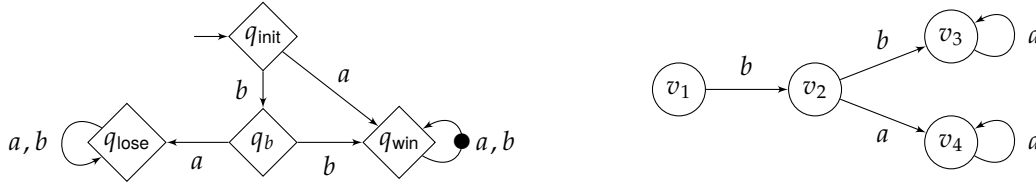
Figure 4.8: Example 4.7.2: DBA recognizing $W = (a + bb)C^\omega$ (left), and arena with no memoryless optimal strategy (right).

of (uniformly) optimal strategies based on $\mathcal{M}$ for both players in finite two-player arenas (Theorem 4.4.1). $\qquad\qquad\square$

Most classical objectives admit the existence of such a w_ε; in particular, all prefix-independent objectives admit it (as any finite word has the same winning continuations as the empty word). The existence of such a w_ε is also weaker than the existence of a *neutral color* [Ohl23], a recent fruitful hypothesis to study the existence of memoryless optimal strategies. Lemma 4.7.1 also led us to find a counterexample when the empty word has no equivalent.

[Ohl23]: Ohlmann (2023), *Characterizing Positionality in Games of Infinite Duration over Infinite Graphs*

> **Example 4.7.2** Let $C = \{a, b\}$ and $W = (a + bb)C^\omega$ be the set of infinite words that either start with a or with bb. This objective is ω-regular and is recognized, e.g., by the DBA in Figure 4.8 (left). Observe that no non-empty word has the same winning continuations as the empty word. This objective does not admit memoryless optimal strategies: in the arena in Figure 4.8 (right), $\mathcal{P}_1$ can win from v_1 by going to v_3 and from v_2 by going to v_4, but there is no uniformly winning strategy from v_1 and v_2.
>
> Yet, in all two-player arenas, $\mathcal{P}_1$ has a memoryless (non-uniformly) winning strategy from every vertex of its winning region. Let $\mathcal{A} = (V, V_1, V_2, E)$ be an arena, and $v \in V$ be a vertex in the winning region of $\mathcal{P}_1$. We show that $\mathcal{P}_1$ has a memoryless winning strategy from v. If $v \in V_1$ and there is an edge $(v, a, v') \in E$, $\mathcal{P}_1$ can simply play this edge. If $v \in V_2$ and all outgoing edges are a-colored, $\mathcal{P}_1$ automatically wins from v. If not, $\mathcal{P}_1$ can win by forcing bb, which can be shown to be doable with a memoryless strategy by distinguishing a few cases.
>
> Through this example, we have distinguished the existence of memoryless (uniformly) optimal strategies for $\mathcal{P}_1$ in two-player arenas from the existence of memoryless non-uniformly winning strategies.

4.8 Further discussion of selected related works

In this section, we give a brief overview of two ways the results from this chapter have been generalized in subsequent works. The first one is a generalization of both the characterization of Theorem 4.4.1 and the one-to-two-player lift (Theorem 4.4.4) to stochastic games and involved this thesis' author [BORV21a]. The second one is a generalization by

[BORV21a]: Bouyer et al. (2021), *Arena-Independent Finite-Memory Determinacy in Stochastic Games*

Kozachinskiy of the one-to-two-player lift to a class of strategies more general than strictly arena-independent ones [Koz22b].

[Koz22b]: Kozachinskiy (2022), *One-To-Two-Player Lifting for Mildly Growing Memory*

4.8.1 Generalization to stochastic games

In [BORV21a], we generalized the one-to-two-player lift to two-player zero-sum *stochastic* games played on finite graphs. In a stochastic game [Sha53], when a player chooses an outgoing edge (often called an *action* in the stochastic case), the following vertex is chosen according to some fixed probability distribution. In deterministic games, when both players fix a strategy, the resulting object is an infinite path in the arena; in stochastic games, it is a *probability distribution* over the set of infinite paths. Therefore, there is an extension of the definition of objectives from this chapter (*preference relations over infinite words*) to preference relations over *distributions* over infinite words. An example of such a general objective with no equivalent in our framework is "reaching a color with probability *exactly* $\frac{1}{2}$" [CFK+12]. In such general stochastic games, the use of *randomization* in strategies may be useful.

[BORV21a]: Bouyer et al. (2021), *Arena-Independent Finite-Memory Determinacy in Stochastic Games*
[Sha53]: Shapley (1953), *Stochastic Games*

[CFK+12]: Chen et al. (2012), *Playing Stochastic Games Precisely*

In general, reasonable extensions of qualitative objectives in the context of stochastic games (for example, maximizing the probability of obtaining an infinite word in the qualitative objective) require more memory than their deterministic counterparts (see, e.g., the *discounted sum with threshold* in [BORV21b, Section 6.2]), hence the need for dedicated tools for stochastic games.

[BORV21b]: Bouyer et al. (2021), *Arena-Independent Finite-Memory Determinacy in Stochastic Games*

The one-to-two-player lift we obtained [BORV21a, Theorem 4.1] is very close to Theorem 4.4.4, and can only be applied to strategies without randomization: if both players have optimal strategies *without randomization* based on a fixed memory structure $\mathcal{M}$ in their respective finite one-player *stochastic* arenas, then the same holds in finite two-player stochastic arenas. Note that a finite one-player stochastic arena is frequently called a *Markov decision process* in the literature.

Technically, we combine insights from this chapter (induction on the set of product arenas) and from an alternative proof for a one-to-two-player lift for memoryless strategies in stochastic games from an unpublished article by Gimbert and Zielonka [GZ09]. This alternative proof is more direct than the one we use here to obtain the lift, but does not provide an equivalence through notions such as $\mathcal{M}$-monotony and $\mathcal{M}$-selectivity. It also allows for a finer quantification on the classes of arenas and of strategies that can be considered; the proof for deterministic arenas is actually a special case.

[GZ09]: Gimbert et al. (2009), *Pure and Stationary Optimal Strategies in Perfect-Information Stochastic Games with Global Preferences*

We finally discuss here the role of randomization in the strategies. In deterministic games, for reasonable (read *Borel*) and determined games [Mar75], one of the players has a (surely) winning strategy with no randomization; in some way, randomization is not needed to win in deterministic games. This result carries over to some extent in stochastic games [Mar98]: if

[Mar75]: Martin (1975), *Borel determinacy*

[Mar98]: Martin (1998), *The Determinacy of Blackwell Games*

the objective consists in maximizing the expected value of a real payoff function $C^\omega \to \mathbb{R}$, then players have ϵ-optimal strategies with no randomization (see [CMJ04, Lemma 10] and [CDGH10, Theorem 4]). Still, we are not considering here memory requirements: even if no randomization is *needed* to play optimally, can randomization help decrease memory usage? The answer is *yes*, and a well-understood case is the one of *Muller conditions* [CdH04; Hor09]. Trade-offs between memory and randomization have also been investigated in other contexts [CRR14; MPR20]. There is therefore a separate direction of study — to which this thesis gives no answer — in trying to understand memory requirements while allowing for some kind of randomization in the strategies.

4.8.2 Generalization to *mildly growing memory*

In [Koz22b], Kozachinskiy looks at the same model of zero-sum deterministic games as in this chapter, and he extends the one-to-two-player lift from this chapter (Theorem 4.4.4) to a class of strategies beyond arena-independent ones, venturing into the lands of (non-arena-independent) finite-memory determinacy.

For our one-to-two-player lift, we required that the amount of memory necessary to play optimally in finite one-player arenas is "constant" in the sense that no matter the size of the finite one-player arena that we consider, there is a fixed memory structure — the size of which does not depend on the arena — that suffices to play optimally. Kozachinskiy relaxes this constraint and allows the size of the memory structures to "grow mildly" with respect to the size of the one-player arenas.

Let $\sqsubseteq$ be a preference relation. Kozachinskiy studies how the size of a sufficient memory structure grows as a function of the size of the one-player arenas. Let $f : \mathbb{N} \setminus \{0\} \to \mathbb{N} \setminus \{0\}$ be a function such that for $n \geq 1$, there is a chromatic memory structure $\mathcal{M}$ with $f(n)$ states that suffices to play optimally for each player in their respective one-player arenas *with at most n vertices*. It turns out that when f is sublinear (it suffices that $\liminf_n \frac{f(n)}{n} = 0$), then $\sqsubseteq$ is finite-memory determined over finite arenas. Moreover, an upper bound on the memory in finite two-player arenas of a given size can be computed as a function of f. In practice, if $f = \mathcal{O}(n^\alpha)$ with $0 < \alpha < 1$, then the amount of memory required to play optimally in finite two-player arenas with n vertices is a function in $\mathcal{O}(n^{\frac{\alpha}{1-\alpha}})$. The proof scheme uses the same technique as [GZ09; BORV21a] discussed above, but pushes it to its limits by making finer observations on the size of all the objects that are manipulated.

Let us compare this result to ours. In the arena-independent case, f can be taken as ultimately constant, so we naturally have that $\liminf_n \frac{f(n)}{n} = 0$, and our result is obtained as a special case. We discussed a counterexample to a general one-to-two-player lift for finite-memory determinacy in Subsection 3.1.2. How to analyze this objective in light of this stronger

[CMJ04]: Chatterjee et al. (2004), *On Nash Equilibria in Stochastic Games*

[CDGH10]: Chatterjee et al. (2010), *Randomness for Free*

[CdH04]: Chatterjee et al. (2004), *Trading Memory for Randomness*

[Hor09]: Horn (2009), *Random Fruits on the Zielonka Tree*

[CRR14]: Chatterjee et al. (2014), *Strategy synthesis for multi-dimensional quantitative objectives*

[MPR20]: Monmege et al. (2020), *Reaching Your Goal Optimally by Playing at Random with No Memory*

[Koz22b]: Kozachinskiy (2022), *One-To-Two-Player Lifting for Mildly Growing Memory*

[GZ09]: Gimbert et al. (2009), *Pure and Stationary Optimal Strategies in Perfect-Information Stochastic Games with Global Preferences*

[BORV21a]: Bouyer et al. (2021), *Arena-Independent Finite-Memory Determinacy in Stochastic Games*

one-to-two-player lift? Clearly, it cannot satisfy the hypothesis of this lift, as it is not finite-memory-determined over finite two-player arenas: the number of states necessary for the memory in one-player arenas cannot be sublinear. On the other hand, Kozachinskiy gave an upper bound: he shows that to play optimally in one-player arenas with $n \geq 1$ vertices, there is a memory structure with $2n + 2$ memory states that suffices. Hence, we cannot hope to have a general one-to-two-player lift for objectives with memory requirements that already grow linearly in finite one-player arenas — we cannot generalize the stronger lift of Kozachinskiy to a class of strategies growing faster than sublinearly.

As far as we know (and as Kozachinskiy knows, as is claimed in his paper), there was no natural example of preference relation from the literature that fits this sublinear hypothesis without also being arena-independent. Kozachinskiy builds one such example to illustrate the applicability of his result [Koz22b, Theorem 5]. We do not define it formally here but discuss it at a high level. This example (actually, a qualitative objective) is defined on $C = \{0, 1\}$ and includes all infinite words with a finite factor 01^t0 for t in some fixed set $T \subseteq \mathbb{N}$. The trick is to take an infinite set T with sufficiently spread out elements (Kozachinskiy takes $T = \{2^{4^k} \subseteq k \in \mathbb{N}\}$). This way, if $t_k < t_{k+1}$ are two consecutive elements in T, for a one-player arenas with n_k vertices such that n_k is significantly greater than t_k but smaller than t_{k+1}, if a player can win, this player can win by counting only up to t_k (thus with a memory structure with about t_k states). By carefully choosing n_k, we then have that $\lim_k \frac{f(n_k)}{n_k} = \lim_k \frac{t_k}{n_k} = 0$, thus $\liminf_n \frac{f(n)}{n} = 0$. By Kozachinskiy's lift, this suffices to show that this objective is finite-memory determined over finite two-player arenas, which was not possible with our result.

Yet, this stronger lift is not an exact frontier of finite-memory determinacy. For instance, the memory requirements of *multi-energy games* are at least linear in the size of the finite one-player arenas, but are still finite in finite two-player arenas [VCD+15; CRR14; JLS15]. This cannot be explained through the stronger lift of Kozachinskiy, as the sublinear hypothesis on the memory of one-player arenas is not satisfied. The quest for a complete understanding of (non-arena-independent) finite-memory determinacy over games played on finite graphs is still ongoing.

[Koz22b]: Kozachinskiy (2022), *One-To-Two-Player Lifting for Mildly Growing Memory*

[VCD+15]: Velner et al. (2015), *The complexity of multi-mean-payoff and multi-energy games*
[CRR14]: Chatterjee et al. (2014), *Strategy synthesis for multi-dimensional quantitative objectives*
[JLS15]: Jurdziński et al. (2015), *Fixed-Dimensional Energy Games are in Pseudo-Polynomial Time*

Characterization of ω-regularity through finite-memory determinacy

<div style="text-align: right; font-size: 3em;">5</div>

Finite-memory determinacy of ω-regular objectives [BL69; GH82; McN93] is a landmark result stating that, in two-player zero-sum turn-based games, finite-memory strategies suffice for both players when the objective is ω-regular (even for games played on graphs of arbitrary cardinality). We refer to Section 1.3 for the significance of this result. We show a reciprocal of that statement: given an objective, when both players can play optimally with a *chromatic* finite memory structure (i.e., whose updates can only observe colors) in all infinite game graphs, then this objective must be ω-regular. This provides a game-theoretic characterization of ω-regular objectives, and this characterization helps obtain memory bounds and automatic representations of these objectives. This result generalizes the work of Colcombet and Niwiński [CN06] discussed in Section 3.2.

Moreover, a by-product of our characterization is a new *one-to-two-player lift*, dealing with another class of game graphs than the one in Chapter 4: to show that chromatic finite-memory structures suffice to play optimally in two-player games on *infinite* graphs, it suffices to show it in the simpler case of *one*-player games on *infinite* graphs. We illustrate our results with the family of discounted-sum objectives, for which ω-regularity depends on the value of some parameters.

The contributions from this chapter are based on joint work with Patricia Bouyer (Université Paris-Saclay, CNRS, ENS Paris-Saclay, LMF) and Mickael Randour (F.R.S.-FNRS & Université de Mons) published in two papers: a conference version in the proceedings of *STACS'22* [BRV22a], and a more extensive journal version in *TheoretiCS* [BRV23].

[BL69]: Büchi et al. (1969), *Definability in the Monadic Second-Order Theory of Successor*
[GH82]: Gurevich et al. (1982), *Trees, Automata, and Games*
[McN93]: McNaughton (1993), *Infinite Games Played on Finite Graphs*

[CN06]: Colcombet et al. (2006), *On the positional determinacy of edge-labeled games*

[BRV22a]: Bouyer et al. (2022), *Characterizing Omega-Regularity Through Finite-Memory Determinacy of Games on Infinite Graphs*

[BRV23]: Bouyer et al. (2023), *Characterizing Omega-Regularity Through Finite-Memory Determinacy of Games on Infinite Graphs*

5.1 Introduction

Right congruence. We recall a notion defined in Section 2.8 that becomes central in this chapter: a well-known tool to study a language L of finite (resp. infinite) words is its *right congruence relation* $\sim_L$: for two finite words w_1 and w_2, we write $w_1 \sim_L w_2$ if for all finite (resp. infinite) words w, $w_1 w \in L$ if and only if $w_2 w \in L$. There is a natural deterministic (potentially infinite) automaton recognizing the equivalence classes for the right congruence, called the *prefix classifier of L* [Sta83; MS97].

The relation between a regular language of *finite* words and its right congruence is given by the Myhill-Nerode theorem [Ner58], which provides a natural bijection between the states of the minimal deterministic automaton recognizing a regular language and the equivalence classes for its right congruence relation. Consequences of this theorem are that (*i*) a language is regular if and only if its right congruence has finitely many equivalence classes, and (*ii*) a regular language can be recognized by a deterministic finite automaton built on top of its prefix classifier.

For the theory of languages of *infinite* words, the situation is not so simple: ω-regular languages have a right congruence with finitely many equivalence classes (Lemma 2.8.8), but having finitely many equivalence classes does not guarantee ω-regularity (for example, a language is *prefix-independent* if and only if its right congruence has exactly one equivalence class, but this does not imply ω-regularity). Moreover, ω-regular languages cannot necessarily be recognized by adding a natural acceptance condition (Büchi, parity, Rabin, Muller...) to their prefix classifier [AF21]. There has been multiple works about the links between a language of infinite words and its prefix classifier; one relevant question, linked to our results, is to understand when a language can be recognized by this prefix classifier [Sta83; MS97; AFS20; BL21; AF21].

Contributions. We characterize the ω-regularity of a language of infinite words W through the strategy complexity of the zero-sum turn-based games on infinite graphs with objective W: the ω-regular languages are *exactly* the chromatic-finite-memory-determined objectives (Theorem 5.4.3). As discussed earlier (Theorem 2.7.11), it is well-known that ω-regular languages admit chromatic-finite-memory optimal strategies [GH82; Mos84; Zie98] — our results yield the other implication. This therefore provides a characterization of ω-regular languages through a game-theoretic and strategic lens.

In this chapter, we talk about *chromatic finite-memory determinacy*, but we recall that for games played on infinite arenas, it is equivalent to arena-independent finite-memory determinacy (Proposition 2.6.14), as used in the previous chapter.

[Sta83]: Staiger (1983), *Finite-State ω-Languages*

[MS97]: Maler et al. (1997), *On Syntactic Congruences for Omega-Languages*

[Ner58]: Nerode (1958), *Linear Automaton Transformations*

[AF21]: Angluin et al. (2021), *Regular ω-languages with an informative right congruence*

[AFS20]: Angluin et al. (2020), *Polynomial Identification of ω-Automata*

[BL21]: Bohn et al. (2021), *Constructing Deterministic ω-Automata from Examples by an Extension of the RPNI Algorithm*

[GH82]: Gurevich et al. (1982), *Trees, Automata, and Games*

[Mos84]: Mostowski (1984), *Regular expressions for infinite trees and a standard form of automata*

[Zie98]: Zielonka (1998), *Infinite Games on Finitely Coloured Graphs with Applications to Automata on Infinite Trees*

Our technical arguments consist in providing a precise connection between the representation of W as a parity automaton and a chromatic memory structure sufficient to play optimally. If strategies based on a chromatic finite-memory structure are sufficient to play optimally for both players, then W is recognized by a parity automaton built on top of the direct product of the *prefix classifier* and this *chromatic memory structure* (Theorem 5.4.1). As sketched in Section 3.2, this result generalizes the work of Colcombet and Niwiński [CN06] in two ways: by relaxing the prefix-independence assumption about the objective, and by generalizing the class of strategies considered from memoryless to chromatic-finite-memory strategies. We recover their result as a special case.

[CN06]: Colcombet et al. (2006), *On the positional determinacy of edge-labeled games*

Moreover, we actually show that chromatic finite-memory determinacy over the *one-player* games of both players is sufficient to show ω-regularity of a language. As ω-regular languages are chromatic-finite-memory-determined over two-player games, we can reduce the problem of chromatic finite-memory determinacy of an objective in *two-player* games to the easier chromatic finite-memory determinacy over *one-player* games (Theorem 5.4.4). We therefore recover a result orthogonal to the main result of Chapter 4 (Theorem 4.4.4) — this time for games played on graphs of arbitrary cardinality — through a different proof technique. By comparison, the proofs of one-to-two-player lifts in finite arenas rely on an *edge-induction technique* (also used in other works about strategy complexity in finite arenas [Kop06; GK14; CD16]) that appears unfit to deal with infinite arenas.

[Kop06]: Kopczyński (2006), *Half-Positional Determinacy of Infinite Games*
[GK14]: Gimbert et al. (2014), *Submixing and Shift-Invariant Stochastic Games*
[CD16]: Chatterjee et al. (2016), *Perfect-Information Stochastic Games with Generalized Mean-Payoff Objectives*

Chapter structure. We start with extra notations used to manipulate memory structures more precisely (Section 5.2). We define notions at the core of our characterization in Section 5.3. Our main results are then discussed in Section 5.4, and their proofs lie in Sections 5.5 and 5.6. We provide an extensive application of our results to discounted-sum objectives, and a brief application to mean-payoff and total-payoff objectives in Section 5.7.

5.2 Preliminaries: manipulating memory structures

Let $\mathcal{M} = (M, m_{\text{init}}, \alpha_{\text{upd}})$ be a memory structure. We introduce extra notations to manipulate memory structures and their transitions throughout this chapter.

We say that a non-empty sequence $\pi = (m_0, c_1) \ldots (m_{k-1}, c_k) \in (M \times C)^+$ is a *path of $\mathcal{M}$ (from m_0 to $\alpha_{\text{upd}}(m_{k-1}, c_k)$)* if for all i such that $1 \leq i \leq k-1$, $m_i = \alpha_{\text{upd}}(m_{i-1}, c_i)$. For convenience, we also consider every element $(m, \_)$ for $m \in M$ and $\_ \notin C$ to be an *empty path of $\mathcal{M}$ (from m to m)*. A non-empty path of $\mathcal{M}$ from m to m' is a *cycle of $\mathcal{M}$ (on m)* if $m = m'$. When

Empty paths of $\mathcal{M}$ are akin to empty arena histories λ_v for arenas, and are also introduced for technical convenience.

we want to emphasize that we work with cycles and not just paths, we usually use letter φ instead of π.

For $\pi = (m_0, c_1) \ldots (m_{k-1}, c_k)$ a path of $\mathcal{M}$, we define $\mathsf{st}(\pi)$ to be the set $\{m_0, \ldots, m_{k-1}\}$, and $\mathsf{col}^*(\pi)$ to be the sequence $c_1 \ldots c_k \in C^*$. For an infinite sequence $(m_0, c_1)(m_1, c_2) \ldots \in (M \times C)^\omega$, we also define $\mathsf{col}^\omega((m_0, c_1)(m_1, c_2) \ldots)$ to be the infinite sequence $c_1 c_2 \ldots \in C^\omega$. If $(m, c) \in M \times C$ occurs in a path π of $\mathcal{M}$, we call (m, c) a *transition of* π and we write abusively $(m, c) \in \pi$.

For $m, m' \in M$, we write $\Pi_{m,m'}$ for the set of paths of $\mathcal{M}$ from m to m', Φ_m for the set of cycles of $\mathcal{M}$ on m, and $\Phi_{\mathcal{M}}$ for the set of all cycles of $\mathcal{M}$ (on any memory state). We extend in a natural way notation col^* to *sets* of paths or cycles of $\mathcal{M}$ to consider their projections to colors (e.g., $\mathsf{col}^*(\Phi_{\mathcal{M}}) = \{\mathsf{col}^*(\varphi) \in C^+ \mid \varphi \in \Phi_{\mathcal{M}}\}$).

In particular, $\mathsf{col}^*(\Pi_{m_1,m_2})$ is corresponds to the previously defined L_{m_1,m_2}.

For $w = c_1 c_2 \ldots \in C^\omega$, we define the *run of $\mathcal{M}$ on w* as the infinite sequence

$$(m_0, c_1)(m_1, c_2) \ldots \in (M \times C)^\omega$$

that w induces in the memory structure ($m_0 = m_{\mathsf{init}}$ and for all $i \geq 1$, $m_i = \alpha_{\mathsf{upd}}(m_{i-1}, c_i)$).

5.3 Concepts

We define two concepts at the core of our characterization. Just like for $\mathcal{M}$-monotony and $\mathcal{M}$-selectivity in Chapter 4, one of them deals with *prefixes* and the other one deals with *cycles*. Let $W \subseteq C^\omega$ be an objective and $\mathcal{M} = (M, m_{\mathsf{init}}, \alpha_{\mathsf{upd}})$ be a memory structure.

Generalizing prefix-independence. We start by defining a natural generalization of prefix-independence to "prefix-independence *modulo* a memory structure". Let $\sim$ be the right congruence of W.

> **Definition 5.3.1** ($\mathcal{M}$-prefix-independence) *Objective W is $\mathcal{M}$-prefix-independent if for all $m \in M$, for all $w_1, w_2 \in \mathsf{col}^*(\Pi_{m_{\mathsf{init}}, m})$, $w_1 \sim w_2$.*

In other words, W is $\mathcal{M}$-prefix-independent if finite words reaching the same state of $\mathcal{M}$ from its initial state have the same winning continuations. The classical notion of *prefix-independence* is equivalent to $\mathcal{M}_{\mathsf{triv}}$-prefix-independence (as all finite words have the exact same set of winning continuations, which is W). If $\sim$ has a finite index, W is in particular $\mathcal{S}_W$-prefix-independent: indeed, two finite words reach the same state of $\mathcal{S}_W$ (if and) only if they are equivalent for $\sim$. Any memory structure $\mathcal{M}$ such that W is $\mathcal{M}$-prefix-independent must have at least one state for each equivalence class for $\sim$ (and thus must be at least as large as $\mathcal{S}_W$), but multiple states may partition the same equivalence class.

We recall that $\mathcal{S}_W$ is the *prefix classifier* of W, defined in Definition 2.8.11 on page 44.
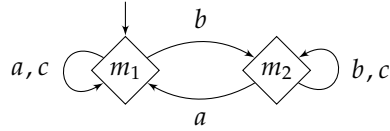
Remark 5.3.2 The $\mathcal{M}$-prefix-independence notion is stronger than $\mathcal{M}$-monotony and $\mathcal{M}$-strong-monotony: we do not just require that finite words reaching the same memory state are comparable in some way w.r.t. their winning continuations (see Subsection 4.3.2), but that they have the exact same winning continuations. This stronger notion will be more suited to deal with games played on *infinite* arenas.

Cycle-consistency.   We move on to our property dealing with cycles: we want to express the property that combining infinitely many "winning cycles" results in a winning word, and similarly for losing cycles and losing words. For $w \in C^*$, we define

$$\Phi_{\mathcal{M}}^{\mathsf{win},w} = \{\varphi \in \Phi_m \mid m = \alpha_{\mathsf{upd}}^*(m_{\mathsf{init}}, w) \text{ and } (\mathsf{col}^*(\varphi))^\omega \in w^{-1}W\}$$

as the cycles on the memory state reached by w in $\mathcal{M}$ that induce winning words when repeated infinitely many times after w. We define

$$\Phi_{\mathcal{M}}^{\mathsf{lose},w} = \{\varphi \in \Phi_m \mid m = \alpha_{\mathsf{upd}}^*(m_{\mathsf{init}}, w) \text{ and } (\mathsf{col}^*(\varphi))^\omega \in w^{-1}\overline{W}\}$$

as their losing counterparts. We emphasize that cycles are not simple and are allowed to go through the same state or transition multiple times.

Definition 5.3.3 ($\mathcal{M}$-cycle-consistency) *Objective W is $\mathcal{M}$-cycle-consistent if for all $w \in C^*$, $(\mathsf{col}^*(\Phi_{\mathcal{M}}^{\mathsf{win},w}))^\omega \subseteq w^{-1}W$ and $(\mathsf{col}^*(\Phi_{\mathcal{M}}^{\mathsf{lose},w}))^\omega \subseteq w^{-1}\overline{W}$.*

What this says is that after any finite word, if we concatenate infinitely many winning (resp. losing) cycles on the memory state reached by that word, then it only produces winning (resp. losing) infinite words.

Example 5.3.4 Let $C = \{a, b, c\}$. Objective $W = \mathsf{Büchi}(a) \cap \mathsf{Büchi}(b)$ is ($\mathcal{M}_{\mathsf{triv}}$-)prefix-independent, but not $\mathcal{M}_{\mathsf{triv}}$-cycle-consistent: for any $w \in C^*$, a and b are both in $\mathsf{col}^*(\Phi_{\mathcal{M}_{\mathsf{triv}}}^{\mathsf{lose},w})$ (as wa^ω and wb^ω are losing), but word $w(ab)^\omega$ is winning. However, W is $\mathcal{M}$-cycle-consistent for the memory structure $\mathcal{M}$ with two states m_1 and m_2 represented in Figure 5.1. For finite words reaching m_1, the losing cycles only see a and c, and combining infinitely many of them gives an infinite word without b, which is a losing continuation of any finite word. The winning cycles are the ones that go to m_2 and then go back to m_1, as they must see both a and b; combining infinitely many of them guarantees a winning continuation after any finite word. A similar reasoning applies to state m_2. Notice that W is also $\mathcal{M}$-prefix-independent. With regard to memory
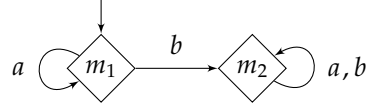
requirements, $\mathcal{M}_{\text{triv}}$ does not suffice to play optimally for $\mathcal{P}_1$ for W, but $\mathcal{M}$ does.

Remark 5.3.5 Just as we compared $\mathcal{M}$-prefix-independence to $\mathcal{M}$-monotony, we may inquire about the links between $\mathcal{M}$-cycle-consistency and $\mathcal{M}$-selectivity.

It is natural that $\mathcal{M}$-selectivity does not imply the seemingly stronger $\mathcal{M}$-cycle-consistency, which may compare cycles in a less structured way. For instance, with $C = \mathbb{Q}$, the mean-payoff objective $\mathsf{MP}^{\geq 0}$ is $\mathcal{M}_{\text{triv}}$-selective, which follows from Proposition 4.5.2 and its memoryless determinacy over finite arenas (Example 3.1.2). Yet, it is not $\mathcal{M}_{\text{triv}}$-cycle-consistent: for all words w in

$$\underbrace{\{0, \dots, 0}_{n \text{ times}}, -1 \in C^+ \mid n \geq 0\},$$

we have that w^ω is losing (w is in $\mathsf{col}^*(\Phi_{\mathcal{M}_{\text{triv}}}^{\text{lose},\varepsilon})$), but there is a way to combine words in this set to produce a winning word (e.g., the word $-1, 0, -1, 0, 0, -1, \dots$).

On the other hand, $\mathcal{M}$-cycle-consistency also does not imply $\mathcal{M}$-selectivity. One detail of $\mathcal{M}$-selectivity that puts a constraint not needed for $\mathcal{M}$-cycle-consistency is that the K_3 in the definition of $\mathcal{M}$-selectivity does not have to be a set of cycles on a memory state. We provide a concrete example.

Let $C = \{a, b\}$ and $W = abC^\omega$. If we consider the memory structure $\mathcal{M}$ in Figure 5.2, then W is $\mathcal{M}$-cycle-consistent: for all finite words w except for ε and a, either all continuations are winning (if $w \in abC^*$) or all continuations are losing. If w is ε or a, then it reaches state m_1 of $\mathcal{M}$, and the only cycles on m_1 are in a^+, are losing, and are losing when infinitely many of them are combined into an infinite word. But W is not $\mathcal{M}$-selective. Indeed, if we take $w = \varepsilon$, $K_1 = \{a\}$, $K_2 = \emptyset$, and $K_3 = b^*$ in the definition of $\mathcal{M}$-selectivity, we have that $[wK_1^*] \cup [wK_2^*] \cup [wK_3] = \{a^\omega, b^\omega\}$ does not contain a winning word, but $[w(K_1 \cup K_2)^* K_3]$ contains the word ab^ω.

In proofs, we will often use a weaker implication of $\mathcal{M}$-cycle-consistency, which is that a finite combination of winning cycles is still a winning cycle (i.e., if $\varphi, \varphi' \in \Phi_{\mathcal{M}}^{\text{win},w}$, then $\varphi\varphi' \in \Phi_{\mathcal{M}}^{\text{win},w}$).

Properties of these concepts. Both $\mathcal{M}$-prefix-independence and $\mathcal{M}$-cycle-consistency hold symmetrically for an objective and its complement, and are stable by product with an arbitrary memory structure (as products generate even smaller sets of prefixes and cycles to consider).

> **Lemma 5.3.6** *Let $W \subseteq C^\omega$ be an objective and $\mathcal{M}$ be a memory structure. Objective W is $\mathcal{M}$-prefix-independent (resp. $\mathcal{M}$-cycle-consistent) if and only if $\overline{W}$ is $\mathcal{M}$-prefix-independent (resp. $\mathcal{M}$-cycle-consistent). If W is $\mathcal{M}$-prefix-independent (resp. $\mathcal{M}$-cycle-consistent), then for all memory structures $\mathcal{M}'$, W is $(\mathcal{M} \otimes \mathcal{M}')$-prefix-independent (resp. $(\mathcal{M} \otimes \mathcal{M}')$-cycle-consistent).*

Proof. We write $\mathcal{M} = (M, m_{\text{init}}, \alpha_{\text{upd}})$.

We assume that W is $\mathcal{M}$-prefix-independent. Thus, for all $m \in M$, for all $w_1, w_2 \in \text{col}^*(\overline{\Pi_{m_{\text{init}}, m}})$, $w_1 \sim w_2$, i.e., $w_1^{-1} W = w_2^{-1} W$. This last equality is equivalent to $\overline{w_1^{-1} W} = \overline{w_2^{-1} W}$, which can be rewritten as $w_1^{-1} \overline{W} = w_2^{-1} \overline{W}$. This shows that $\overline{W}$ is $\mathcal{M}$-prefix-independent.

To show that W is $\mathcal{M}$-cycle-consistent if and only if $\overline{W}$ is $\mathcal{M}$-cycle-consistent, notice that the winning cycles for W are exactly the losing cycles for $\overline{W}$, and vice versa.

Let $\mathcal{M}' = (M', m'_{\text{init}}, \alpha'_{\text{upd}})$ be a memory structure. We assume that W is $\mathcal{M}$-prefix-independent and we show that W is $(\mathcal{M} \otimes \mathcal{M}')$-prefix-independent. The sets of prefixes to consider are smaller in $\mathcal{M} \otimes \mathcal{M}'$ than in $\mathcal{M}$: for all $(m, m') \in M \times M'$, $\text{col}^*(\Pi_{(m_{\text{init}}, m'_{\text{init}}), (m, m')}) \subseteq \text{col}^*(\Pi_{m_{\text{init}}, m})$. Therefore, for all $w_1, w_2 \in \text{col}^*(\Pi_{(m_{\text{init}}, m'_{\text{init}}), (m, m')})$, we also have $w_1, w_2 \in \text{col}^*(\Pi_{m_{\text{init}}, m})$, so by $\mathcal{M}$-prefix-independence, $w_1 \sim w_2$.

We now assume that W is $\mathcal{M}$-cycle-consistent and we show that W is $(\mathcal{M} \otimes \mathcal{M}')$-cycle-consistent. The sets of winning and losing cycles to consider are smaller in $\mathcal{M} \otimes \mathcal{M}'$ than in $\mathcal{M}$: for all $w \in C^*$, $\text{col}^*(\Phi_{\mathcal{M} \otimes \mathcal{M}'}^{\text{win}, w}) \subseteq \text{col}^*(\Phi_{\mathcal{M}}^{\text{win}, w})$ and $\text{col}^*(\Phi_{\mathcal{M} \otimes \mathcal{M}'}^{\text{lose}, w}) \subseteq \text{col}^*(\Phi_{\mathcal{M}}^{\text{lose}, w})$. By $\mathcal{M}$-cycle-consistency, for all $w \in C^*$, we have $(\text{col}^*(\Phi_{\mathcal{M}}^{\text{win}, w}))^\omega \subseteq w^{-1} W$ and $(\text{col}^*(\Phi_{\mathcal{M}}^{\text{lose}, w}))^\omega \subseteq w^{-1} \overline{W}$, so we also have $(\text{col}^*(\Phi_{\mathcal{M} \otimes \mathcal{M}'}^{\text{win}, w}))^\omega \subseteq w^{-1} W$ and $(\text{col}^*(\Phi_{\mathcal{M} \otimes \mathcal{M}'}^{\text{lose}, w}))^\omega \subseteq w^{-1} \overline{W}$. □

An interesting property of languages defined by a parity automaton is that they satisfy both aforementioned concepts with their underlying automaton structure.

> **Lemma 5.3.7** *Let $\mathcal{P} = (\mathcal{S}, p)$ be a parity automaton and $W = \mathcal{L}(\mathcal{P})$ be the objective recognized by $\mathcal{P}$. Objective W is $\mathcal{S}$-prefix-independent and $\mathcal{S}$-cycle-consistent.*

> **Remark 5.3.8** Note that in a slight abuse of notations, we allow in this chapter to use automaton structures (on alphabet C) as memory structures, as in the statement of the above lemma. We recall that automaton structures on alphabet C are tuples $\mathcal{S} = (Q, C, q_{\text{init}}, \delta)$ with finite Q, $q_{\text{init}} \in Q$, and $\delta \colon Q \times C \to Q$, and that memory structures are tuples $\mathcal{M} = (M, m_{\text{init}}, \alpha_{\text{upd}})$ with finite M, $m_{\text{init}} \in M$, and $\alpha_{\text{upd}} \colon M \times C \to M$. They are usually used for different purposes, but this chapter will blur the line between both objects and consider both automaton structures

as memory structures and memory structures as automaton structures.

Proof of Lemma 5.3.7. By definition of the parity acceptance condition, any two finite words reaching the same state of the underlying structure have the same winning continuations (Lemma 2.8.7). Therefore, W is $\mathcal{S}$-prefix-independent.

Also, the winning (resp. losing) cycles of $\mathcal{S}$ after any finite word are exactly the ones that have an even (resp. odd) maximal priority. Therefore, combining infinitely many winning (resp. losing) cycles can only produce a winning (resp. losing) infinite word. □

5.4 Characterization

We state our main technical result, which will be the main tool to derive more elegant theoretical results.

Theorem 5.4.1 *Let $W \subseteq C^\omega$ be an objective, $\sim$ be its right congruence, and $\mathcal{M}$ be a memory structure.*

1. *If $\mathcal{M}$ suffices for both players for W in their respective countable one-player arenas, then $\sim$ has a finite index (in particular, $\mathcal{S}_W$ is well-defined and W is $\mathcal{S}_W$-prefix-independent) and W is $\mathcal{M}$-cycle-consistent.*
2. *If W is $\mathcal{M}$-prefix-independent and $\mathcal{M}$-cycle-consistent, then W is ω-regular and can be recognized by a deterministic parity automaton built on top of $\mathcal{M}$.*

To obtain a memory structure $\mathcal{M}'$ such that W is both $\mathcal{M}'$-prefix-independent and $\mathcal{M}'$-cycle-consistent from the first item, we can take $\mathcal{M}' = \mathcal{S}_W \otimes \mathcal{M}$ by Lemma 5.3.6.

We prove the two items of this theorem respectively in Sections 5.5 and 5.6. Before detailing the consequences of this result, we show a few examples.

Example 5.4.2 We consider four objectives, and discuss in each case their prefix classifier and a minimal memory structure (w.r.t. the number of states) sufficient to play optimally for both players. They are represented in Figure 5.3.

▸ Let W_1 be a parity condition ($C = \{0, \ldots, n\}$). It is prefix-independent ($\mathcal{S}_{W_1} = \mathcal{M}_{\mathsf{triv}}$) and memoryless-determined ($\mathcal{M}_{\mathsf{triv}}$ is a minimal memory structure). By Theorem 5.4.1, W_1 is $\mathcal{M}_{\mathsf{triv}}$-prefix-independent and $\mathcal{M}_{\mathsf{triv}}$-cycle-consistent. Just from these last two facts, we obtain that W_1 has to be recognizable by a parity automaton built on top of $\mathcal{M}_{\mathsf{triv}}$, and therefore has to be a parity condition. This provides a characterization of parity conditions, and is exactly Colcombet and Niwiński's result (see Theorem 3.2.2) [CN06].

▸ Let $C = \{a, b\}$ and $W_2 = b^*ab^*aC^\omega$ be the language of infinite words with at least two occurrences of a. The prefix classifier $\mathcal{S}_{W_2}$

[CN06]: Colcombet et al. (2006), *On the positional determinacy of edge-labeled games*

Objective	Prefix classifier $\mathcal{S}_W$	Memory $\mathcal{M}$
$C = \{0, \ldots, n\}$, Parity condition	$\longrightarrow \diamondsuit \circlearrowright C$	$\longrightarrow \diamondsuit \circlearrowright C \mapsto \{0, \ldots, n\}$
$C = \{a, b\}$, $W_2 = b^* a b^* a C^\omega$	$\overset{b\,\mid\,1}{\longrightarrow \diamondsuit} \overset{a\,\mid\,1}{\longrightarrow} \overset{b\,\mid\,1}{\diamondsuit} \overset{a\,\mid\,1}{\longrightarrow} \diamondsuit \circlearrowright C \mid 2$	$\longrightarrow \diamondsuit \circlearrowright C$
$C = \{a, b\}$, $W_3 = C^*(ab)^\omega$	$\longrightarrow \diamondsuit \circlearrowright C$	$b \mid 1 \; \diamondsuit \underset{b\,\mid\,0}{\overset{a\,\mid\,0}{\rightleftarrows}} \diamondsuit \; a \mid 1$
$C = \mathbb{Q}$, $W_4 = \mathsf{MP}^{\geq 0}$	$\longrightarrow \diamondsuit \circlearrowright C$	No finite structure

Figure 5.3: Four objectives (the first three are ω-regular), their prefix classifier, a minimal memory structure for each of them, and a way (in red) to realize the first three as parity automata built on top of these structures. We recall that a transition of a parity automaton from q to q' with label $c \mid k$ means that $\delta(q, c) = q'$ and $p(q, c) = k$.

of this objective has three states that count the occurrences of a (0, 1, or ≥ 2), and W_2 is memoryless-determined (we do not prove it here, but it was for instance shown in [BFMM11, Lemma 13] and it follows easily from the results of the subsequent Chapter 7). Hence, as properties are stable by product (Lemma 5.3.6), W_2 is both $\mathcal{S}_{W_2}$-prefix-independent and $\mathcal{S}_{W_2}$-cycle-consistent. Thus, Theorem 5.4.1 tells us that W_2 can be recognized by a parity automaton built on top of $\mathcal{S}_{W_2}$ by carefully labeling its transitions with adequate priorities, which we do in Figure 5.3.

▶ Let $C = \{a, b\}$ and $W_3 = C^*(ab)^\omega$ be the language of words ending with $(ab)^\omega$. This objective is prefix-independent ($\mathcal{S}_{W_3} = \mathcal{M}_{\text{triv}}$) and a memory structure $\mathcal{M}$ with two states suffices to play optimally for both players (we simply need to know whether a or b was just seen). Repeating the arguments above, objective W_3 is $\mathcal{M}$-prefix-independent and $\mathcal{M}$-cycle-consistent, so it admits a representation using a parity automaton built on top of $\mathcal{M}$. We depict such a representation in Figure 5.3.

▶ Let $C = \mathbb{Q}$ and $W_4 = \mathsf{MP}^{\geq 0}$. This objective is prefix-independent ($\mathcal{S}_{W_4} = \mathcal{M}_{\text{triv}}$). As W_4 is not ω-regular, it has no hope to admit a sufficient finite memory structure, as Theorem 5.4.1 would then imply that $\mathsf{MP}^{\geq 0}$ is recognizable by a parity automaton built on top of that structure. There is therefore no sufficient finite memory structure, which we already knew by exhibiting an infinite arena requiring infinite memory in Example 3.2.1.

Notice that in all of these examples, for simplicity, at least one structure among the prefix classifier and the minimal memory structure is trivial.

In more complicated examples, both structures may be non-trivial. One such example can be generated from the examples above: consider objective $W_2 \cup W_3$ on colors $\{a, b, c, d\}$ after renaming the colors used to define W_3 to c and d.

Echoing Remark 2.7.5, observe that Theorem 5.4.1 would not be true if we considered automata with acceptance conditions defined on the automaton *states* (rather than transitions, as we do here). For instance, W_3 in the example above needs three states to be represented as a state-based DPA, so the product of the prefix classifier and a memory structure would not suffice. It is likely that Theorem 5.4.1 can be rephrased using state-based DPAs, but this would require an additional (and less elegant) blow-up in the state space.

A state-based DPA with three states recognizing W_3 is

We now discuss multiple consequences of Theorem 5.4.1 advertised in the introduction, including notably a strategic characterization of ω-regular languages and a novel one-to-two-player-lift.

Characterizing ω-regularity. The first one is a characterization of ω-regularity through a strategic property.

> **Theorem 5.4.3** (Strategic characterization of ω-regularity) *Let $W \subseteq C^\omega$ be a language of infinite words. Language W is ω-regular if and only if it is chromatic-finite-memory-determined (over arenas of arbitrary cardinality).*

Proof. One implication is well-known (and was already stated in Theorem 2.7.11): if W is ω-regular, then it can be recognized by a deterministic parity automaton whose underlying structure can be used as a memory that suffices to play optimally for both players, in arenas of any cardinality (see Theorem 2.7.11 [Mos84; EJ91; Mos91; Zie98]). For the other direction, if W is chromatic-finite-memory-determined, then there exists a memory structure $\mathcal{M}$ such that $\mathcal{M}$ suffices to play optimally for both players for W. In particular, $\mathcal{M}$ suffices for both players in their countable one-player arenas, so by Theorem 5.4.1 (first item), $\sim$ has a finite index and W is $\mathcal{M}$-cycle-consistent. In particular, by Lemma 5.3.6, W is $(\mathcal{S}_W \otimes \mathcal{M})$-prefix-independent and $(\mathcal{S}_W \otimes \mathcal{M})$-cycle-consistent, so by Theorem 5.4.1 (second item), W is ω-regular and can be recognized by a deterministic parity automaton built on top of $\mathcal{S}_W \otimes \mathcal{M}$. $\qquad\square$

One-to-two-player lift. The second one is a novel one-to-two-player lift, orthogonal to the one in Chapter 4.

> **Theorem 5.4.4** (One-to-two-player lift, infinite arenas) *Let $W \subseteq C^\omega$ be an objective. If a memory structure $\mathcal{M}$ suffices to play optimally for both players in their respective one-player arenas, then $\mathcal{S}_W \otimes \mathcal{M}$ suffices to play optimally*
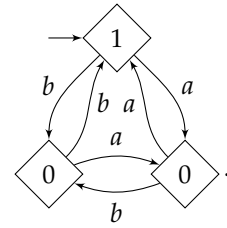
We considered the syntactically stronger *arena-independent* finite-memory determinacy in the previous chapter (about games played on finite arenas), but we recall that chromatic and arena-independent finite-memory determinacy coincide for games played on infinite arenas (Proposition 2.6.14). In this chapter, we prefer the use of *chromatic* finite-memory determinacy in our statements.

[Mos84]: Mostowski (1984), *Regular expressions for infinite trees and a standard form of automata*
[EJ91]: Emerson et al. (1991), *Tree Automata, Mu-Calculus and Determinacy (Extended Abstract)*
[Mos91]: Mostowski (1991), *Games with Forbidden Positions*
[Zie98]: Zielonka (1998), *Infinite Games on Finitely Coloured Graphs with Applications to Automata on Infinite Trees*

for both players (in all arenas).

Proof. By Theorem 5.4.1: if $\mathcal{M}$ suffices in one-player arenas, then $\sim$ has a finite index and W is $\mathcal{M}$-cycle-consistent. Again by Lemma 5.3.6 and Theorem 5.4.1, as W can be recognized by a parity automaton built on top of $\mathcal{S}_W \otimes \mathcal{M}$, W is determined and strategies based on $\mathcal{S}_W \otimes \mathcal{M}$ suffice to play optimally in all two-player arenas. □

We observe that this one-to-two-player lift is less "tight" than the one for finite arenas (Chapter 4, Theorem 4.4.4). Indeed, there is a small blow-up of size $\mathcal{S}_W$ when going from memory requirements of an objective W in one-player arenas to memory requirements in two-player arenas. This blow-up is necessary for our proof technique, as we go through the representation of an objective using a DPA, and taking the product with $\mathcal{S}_W$ is necessary for this purpose in general (see for instance objective $W_2 = b^*ab^*aC^\omega$ in Example 5.4.2). However, if we ignore the representation of the objective as a DPA and we speak solely about memory requirements, we do not know whether this blow-up is necessary. This suggests the following conjecture.

> **Conjecture 5.4.5** (Tight one-to-two-player lift, infinite arenas) *Let $W \subseteq C^\omega$ be an ω-regular objective. If a memory structure $\mathcal{M}$ suffices to play optimally for both players in their respective one-player arenas, then $\mathcal{M}$ suffices to play optimally for both players (in all arenas).*

Throughout the rest of the thesis, we will see two specific classes of ω-regular objectives in which this conjecture is shown to hold: the case of *regular* objectives (Chapter 7) and, for memoryless strategies, the case of objectives recognizable by deterministic Büchi automata (Chapter 8). The proofs are very different from the techniques in this chapter and are by-products of precise characterizations of the memory requirements for these objectives.

One third class of objectives in which the conjecture holds, which is easily derived from results in this chapter, is the one of prefix-independent objectives. If an objective W is prefix-independent (i.e., $\sim$ has index 1 and $\mathcal{S}_W = \mathcal{M}_{\text{triv}}$), and memory structure $\mathcal{M}$ suffices to play optimally in one-player games, then W is recognized by a parity automaton built on top of $\mathcal{M}_{\text{triv}} \otimes \mathcal{M}$, which is isomorphic to $\mathcal{M}$. This implies that the exact same memory $\mathcal{M}$ can be used by both players to play optimally in two-player arenas, with no increase in memory. Conjecture 5.4.5 is automatically proved for prefix-independent objectives.

If, additionally, $\mathcal{M} = \mathcal{M}_{\text{triv}}$ (i.e., memoryless strategies suffice to play optimally in one-player arenas), we recover exactly the result from Colcombet and Niwiński [CN06]: W can be recognized by a parity automaton built on top of $\mathcal{M}_{\text{triv}}$, so we can directly assign a priority to each color with a

Actually, an even stronger version of this lift holds for these two examples, in which we do not have to combine the memory structures of both players.

[CN06]: Colcombet et al. (2006), *On the positional determinacy of edge-labeled games*

function $p \colon C \to \{0, \ldots, n\}$ such that an infinite word $w = c_1 c_2 \ldots \in C^\omega$ is in W if and only if $\limsup_{i \to \infty} p(c_i)$ is even.

Recognizability by the prefix classifier. An interesting property of some ω-regular languages is that they can be *recognized by building an acceptance condition on top of their prefix classifier* [Sta83; MS97], which is a helpful property for the learning of languages [MP95; AFS20; BL21; AF21]. Here, for the parity acceptance condition, we have a strategic characterization of this property: Theorem 5.4.1 entails that W can be recognized by building a (transition-based) DPA on top of the prefix classifier $\mathcal{S}_W$ if and only if $\mathcal{S}_W$ suffices to play optimally for both players for W (and, more precisely, if and only if W is $\mathcal{S}_W$-cycle-consistent). Existing works trying to find the boundaries of languages recognizable by their prefix classifier consider multiple acceptance conditions [Sta83; MS97; AFS20; AF21], but not the "transition-based parity" one, so our result is not directly comparable to theirs.

> **Corollary 5.4.6** *Let $W \subseteq C^\omega$ be an ω-regular objective and $\mathcal{S}_W$ be its prefix classifier. The following are equivalent:*
>
> 1. *W is recognized by a DPA built on top of $\mathcal{S}_W$;*
> 2. *$\mathcal{S}_W$ suffices to play optimally for both players;*
> 3. *W is $\mathcal{S}_W$-cycle-consistent.*

Proof. Implication 1. $\implies$ 2. follows from the memoryless determinacy of parity games (Theorem 2.6.3 [EJ91; Mos91]). Implication 2. $\implies$ 3. follows from the first item of Theorem 5.4.1. Implication 3. $\implies$ 1. follows from the second item of Theorem 5.4.1: we have by definition that W is $\mathcal{S}_W$-prefix-independent, so if it is additionally $\mathcal{S}_W$-cycle-consistent, then W can be recognized by a parity automaton built on top of $\mathcal{S}_W$. $\qquad\square$

From countable to arbitrary arenas. There is one specificity of Theorem 5.4.1 that we have not exploited yet: a hypothesis about *countable* (one-player) arenas suffices to obtain the result. This implies the following.

> **Corollary 5.4.7** (Countable-to-arbitrary lift) *Let $W \subseteq C^\omega$ be an objective. Objective W is chromatic-finite-memory-determined over countable arenas if and only if it is chromatic-finite-memory-determined (in arenas of any cardinality).*

Proof. If objective W is chromatic-finite-memory-determined over countable arenas, there is a memory structure $\mathcal{M}$ that suffices to play optimally in all countable arenas (in particular, $\mathcal{M}$ suffices in countable one-player arenas). By using the two items of Theorem 5.4.1 sequentially, we obtain as in the above proofs that $\mathcal{S}_W \otimes \mathcal{M}$ suffices to play optimally in all arenas

[Sta83]: Staiger (1983), *Finite-State ω-Languages*

[MS97]: Maler et al. (1997), *On Syntactic Congruences for Omega-Languages*

This property is often called *having an informative right congruence* in the literature.

[MP95]: Maler et al. (1995), *On the Learnability of Infinitary Regular Sets*

[AFS20]: Angluin et al. (2020), *Polynomial Identification of ω-Automata*

[BL21]: Bohn et al. (2021), *Constructing Deterministic ω-Automata from Examples by an Extension of the RPNI Algorithm*

[AF21]: Angluin et al. (2021), *Regular ω-languages with an informative right congruence*

[EJ91]: Emerson et al. (1991), *Tree Automata, Mu-Calculus and Determinacy (Extended Abstract)*

[Mos91]: Mostowski (1991), *Games with Forbidden Positions*

(of arbitrary cardinality). In particular, W is chromatic-finite-memory-determined. □

We showed that there was a gap between the memory requirements in finite and in countable arenas (Example 3.2.1), but chromatic finite-memory determinacy over countable arenas extends to all arenas of any cardinality.

Although the setting is different, this contrasts with *stochastic* games for which, in some contexts, simple strategies may suffice in countable arenas but not in uncountable arenas [Orn69].

[Orn69]: Ornstein (1969), *On the Existence of Stationary Optimal Strategies*

Classes of arenas. We discuss how much Theorem 5.4.1 depends upon our model of arenas.

Multiple objectives are chromatic-finite-memory-determined over finite arenas (finitely many vertices and edges), but are not over infinite arenas. A few examples are discounted-sum games [Sha53], mean-payoff games [EM79], total-payoff games [GZ04], energy (also called *one-counter*) games [BBE10] which are all memoryless-determined over finite arenas but require infinite memory to play optimally in some infinite arenas (we discuss some of these in Section 5.7). In particular, Theorem 5.4.3 tells us that the derived objectives are not ω-regular.

[Sha53]: Shapley (1953), *Stochastic Games*

[EM79]: Ehrenfeucht et al. (1979), *Positional Strategies for Mean Payoff Games*

[GZ04]: Gimbert et al. (2004), *When Can You Play Positionally?*

[BBE10]: Brázdil et al. (2010), *One-Counter Stochastic Games*

[GW06]: Grädel et al. (2006), *Positional Determinacy of Games with Infinitely Many Priorities*

[CN06]: Colcombet et al. (2006), *On the positional determinacy of edge-labeled games*

Strangely, the fact that our arenas have colors on *edges* and not on *vertices* is crucial for the result. Indeed, there exists an objective (a generalization of a parity condition with infinitely many priorities [GW06]) that is memoryless-determined over vertex-labeled infinite arenas, but not in edge-labeled infinite arenas (as we consider here). This particularity was already discussed [CN06], and it was also shown that the same objective is memoryless-determined over edge-labeled arenas with finite branching. Therefore, the fact that we allow *infinite branching* in our arenas is also necessary for Theorem 5.4.3. Another example of an objective with finite memory requirements in finitely branching arenas for one player but infinite memory requirements in infinitely branching arenas is presented in [CFH14, Section 4].

[CFH14]: Colcombet et al. (2014), *Playing Safe*

In the upcoming Sections 5.5 and 5.6, we prove respectively the first item and the second item of Theorem 5.4.1.

5.5 Two properties of chromatic finite-memory determinacy

Let $W \subseteq C^\omega$ be an objective, $\sim$ be the right congruence of W, $\preceq$ be the prefix preorder of W, and $\mathcal{M} = (M, m_{\text{init}}, \alpha_{\text{upd}})$ be a memory structure, fixed for this section. We aim to show the first item of Theorem 5.4.1, which is that for a memory structure $\mathcal{M}$, the sufficiency of $\mathcal{M}$ in countable
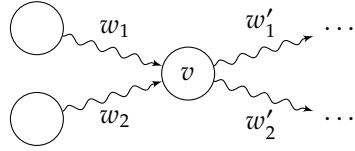
one-player arenas to play optimally for both players implies that $\sim$ has a finite index and that W is $\mathcal{M}$-cycle-consistent.

Finite index of $\sim$. We will use preorder $\preceq$ to deduce that $\sim$ has a finite index by showing that under hypotheses about the optimality of strategies based on $\mathcal{M}$ in countable one-player arenas, (*i*) on each subset $\mathsf{col}^*(\Pi_{m_{\text{init}},m})$ of C^* for $m \in M$, preorder $\preceq$ is total (Lemma 5.5.1) (*ii*) on each subset $\mathsf{col}^*(\Pi_{m_{\text{init}},m})$ of C^* for $m \in M$, preorder $\preceq$ has no infinite increasing nor decreasing sequence (Lemma 5.5.2).

We recall that for $w_1, w_2 \in C^*$, $w_1 \preceq w_2$ if $w_1^{-1}W \subseteq w_2^{-1}W$ (meaning that any continuation that is winning after w_1 is also winning after w_2).

> **Lemma 5.5.1** *Assume $\mathcal{P}_1$ has optimal strategies based on $\mathcal{M}$ on all its countable one-player arenas. Then, for all $m \in M$, preorder $\preceq$ is total on $\mathsf{col}^*(\Pi_{m_{\text{init}},m})$.*

Proof. Let $m \in M$. Let $w_1, w_2 \in \mathsf{col}^*(\Pi_{m_{\text{init}},m})$; we show that $w_1 \not\preceq w_2$ implies $w_2 \preceq w_1$. If $w_1 \not\preceq w_2$, then there exists $w_1' \in C^\omega$ such that $w_1 w_1' \in W$ and $w_2 w_1' \notin W$. We show that $w_2 \preceq w_1$, i.e., that $w_2^{-1}W \subseteq w_1^{-1}W$. Let $w_2' \in w_2^{-1}W$. We build a countable one-player arena of $\mathcal{P}_1$, depicted in Figure 5.4, that merges the ends of finite chains for w_1 and w_2 and the starts of the infinite chains for w_1' and for w_2' in a vertex v.

The construction in this proof is very similar to the one of Proposition 4.5.1, except that we compare finite words w.r.t. arbitrary continuations (which requires an infinite arena in general), and not just regular ones. We also permit ourselves to go a bit faster, as a similar and more complex construction has already been fully detailed.

It is possible to win after seeing w_1 or w_2, by choosing respectively w_1' or w_2' in the merged vertex v. Moreover, there must be a strategy based on $\mathcal{M}$ that wins from the starts of the chains of both w_1 and w_2, which means that in both cases the same choice has to be made in v (as memory state m is reached in both cases). Continuing to w_1' in v would be losing after w_2, so w_2' must be winning after w_1. Therefore, $w_2' \in w_1^{-1}W$. $\quad\square$

> **Lemma 5.5.2** *Assume $\mathcal{P}_1$ has optimal strategies based on $\mathcal{M}$ in all its countable one-player arenas. For all $m \in M$, there is no infinite decreasing sequence of finite words for $\preceq$ in $\mathsf{col}^*(\Pi_{m_{\text{init}},m})$.*

Proof. Let $m \in M$. Assume by contraposition that there is an infinite decreasing sequence of finite words $w_1 \succ w_2 \succ w_3 \succ \ldots$, with $w_i \in \mathsf{col}^*(\Pi_{m_{\text{init}},m})$ for $i \geq 1$. Then for all $i \geq 1$, there exists $w_i' \in C^\omega$ such that $w_i w_i' \in W$ and $w_{i+1} w_i' \notin W$. We create a countable one-player arena of $\mathcal{P}_1$ in which we merge the ends of chains for all w_i to the starts of chains for all w_i' in a vertex v, for all $i \geq 1$. This arena looks like the one in Figure 5.4, but with infinitely many finite words going in v and infinitely many infinite words going out of v. In this arena, for all $i \geq 1$, it is possible to win from the start of the chain for w_i, but there is no strategy based on $\mathcal{M}$ winning from all the starts of the chains simultaneously. Therefore,

$\mathcal{M}$ is not sufficient to play optimally in all countable one-player arenas of $\mathcal{P}_1$. □

We will also use this last lemma from the point of view of $\mathcal{P}_2$. If consider preorder $\preceq_{\overline{W}}$ (i.e., the prefix preorder for $\mathcal{P}_2$), we obtain $w_1 \preceq_{\overline{W}} w_2$ if and only if $w_2 \preceq_W w_1$ because for any finite word $w \in C^*$, $w^{-1}\overline{W} = \overline{w^{-1}W}$.

We can now combine the results of Lemmas 5.5.1 and 5.5.2 to find that $\sim$ has a finite index if $\mathcal{M}$ suffices to play optimally in countable one-player arenas.

> **Proposition 5.5.3** (Necessity of finite index) *If both $\mathcal{P}_1$ and $\mathcal{P}_2$ have optimal strategies based on $\mathcal{M}$ in their countable one-player arenas, then the right congruence $\sim$ has a finite index.*

Proof. Using Lemma 5.5.2 along with the hypothesis about $\mathcal{P}_1$, we have that for all $m \in M$, there are no infinite decreasing sequence of words in $\mathsf{col}^*(\Pi_{m_{\mathsf{init}},m})$ for $\preceq$. Using the same result replacing $\mathcal{P}_1$ with $\mathcal{P}_2$, we obtain that there is no infinite decreasing sequence for $\preceq_{\overline{W}}$, or in other words, that there is no infinite increasing sequence for $\preceq$. For $m \in M$, as $\preceq$ is total in $\mathsf{col}^*(\Pi_{m_{\mathsf{init}},m})$ (Lemma 5.5.1), we conclude that there are only finitely many equivalence classes for $\sim$ in $\mathsf{col}^*(\Pi_{m_{\mathsf{init}},m})$. As M is finite, there are only finitely many equivalence classes for $\sim$ in $\bigcup_{m \in M} \mathsf{col}^*(\Pi_{m_{\mathsf{init}},m}) = C^*$. □

Under the existence of a memory structure sufficient to play optimally for W in countable one-player arenas, we can therefore consider the prefix classifier $\mathcal{S}_W$ of W.

> **Remark 5.5.4** As in Chapter 4 for the close $\mathcal{M}$-monotony notion, we used the existence of *uniformly* winning strategies in *one*-player arenas in order to prove the above lemmas. As discussed in Section 4.7, we could modify this hypothesis to the existence of (*non-necessarily uniformly*) winning strategies in *two*-player arenas and obtain the same results, if there exists a non-empty word w_ε with the same continuations as the empty word.
> Exactly as in Lemma 4.7.1, we could show that uniformity is for free for chromatic finite-memory determinacy over infinite arenas, under the existence of such a word w_ε.

$\mathcal{M}$**-cycle-consistency of** W. We now prove in a straightforward way that the sufficiency of $\mathcal{M}$ in countable one-player arenas implies $\mathcal{M}$-cycle-consistency of W.

> **Proposition 5.5.5** (Necessity of $\mathcal{M}$-cycle-consistency) *If both $\mathcal{P}_1$ and $\mathcal{P}_2$ have optimal strategies based on $\mathcal{M}$ in their countable one-player arenas, then objective W is $\mathcal{M}$-cycle-consistent.*
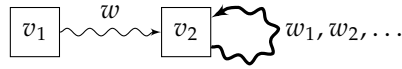
Proof. Let $w \in C^*$ and $m = \alpha_{\mathsf{upd}}^*(m_{\mathsf{init}}, w)$. We show that $(\mathsf{col}^*(\Phi_{\mathcal{M}}^{\mathsf{win},w}))^\omega \subseteq w^{-1}W$. If $\mathsf{col}^*(\Phi_{\mathcal{M}}^{\mathsf{win},w})$ is empty, this is true. If not, let $w_1, w_2, \ldots$ be an infinite sequence of finite words in $\mathsf{col}^*(\Phi_{\mathcal{M}}^{\mathsf{win},w})$ — we show that the infinite word $w_1 w_2 \ldots$ is in $w^{-1}W$. We consider the countable one-player arena of $\mathcal{P}_2$ depicted in Figure 5.5: it starts with a chain for w from a vertex v_1 to a vertex v_2, and v_2 offers a choice among cycles for each finite word in $\{w_1, w_2, \ldots\}$. In this arena, $\mathcal{P}_2$ has no winning strategy based on $\mathcal{M}$ from v_1, since the same memory state m is always reached in v_2 (hence the same choice must always be made in v_2), and repeating any cycle in $\mathsf{col}^*(\Phi_{\mathcal{M}}^{\mathsf{win},w})$ forever after w is winning for $\mathcal{P}_1$ by definition of $\mathsf{col}^*(\Phi_{\mathcal{M}}^{\mathsf{win},w})$. Therefore, $\mathcal{P}_2$ also has no winning strategy at all, which means in particular that the infinite word $w_1 w_2 \ldots$ must be a winning continuation of w. Hence, $w_1 w_2 \ldots$ is in $w^{-1}W$. We get that $(\mathsf{col}^*(\Phi_{\mathcal{M}}^{\mathsf{win},w}))^\omega \subseteq w^{-1}W$.

Using a similar one-player arena of $\mathcal{P}_1$, we can show in a symmetric way that $(\mathsf{col}^*(\Phi_{\mathcal{M}}^{\mathsf{lose},w}))^\omega \subseteq w^{-1}\overline{W}$ for all $w \in C^*$. $\qquad\square$

The reciprocal of this result is false, as shown in the following example.

Example 5.5.6 We discuss again the objective from Remark 5.3.5: let $C = \{a, b\}$ and $W = abC^\omega$. We consider the same memory structure $\mathcal{M}$ in Figure 5.6 (left); we already argued that W is $\mathcal{M}$-cycle-consistent. Yet, this structure does not suffice to play optimally in arena $\mathcal{A}$ in Figure 5.6 (center), as seeing a does not change the memory state.

Notice that the prefix classifier $\mathcal{S}_W$, in Figure 5.6 (right), has four states (corresponding to equivalence classes $[\varepsilon]_\sim$, $[a]_\sim$, $[ab]_\sim$, and $[b]_\sim$) and suffices to play optimally.

Remark 5.5.7 As discussed in Section 5.4, Theorem 5.4.1 does not hold if we assume chromatic finite-memory determinacy over arenas in which *vertices* rather than edges are labeled with colors. Proposition 5.5.5 is an example of a step in the proof of Theorem 5.4.1 that would not work
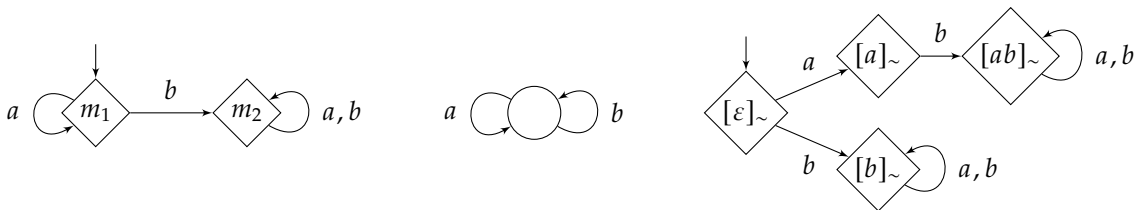

Figure 5.6: Memory structure $\mathcal{M}$ (left), arena $\mathcal{A}$ (center), and prefix classifier $\mathcal{S}_W$ (right) used in Example 5.5.6.

with vertex-labeled arenas: the construction in Figure 5.5 is not possible (there would have to be a color labeling v_2 seen at the start of every cycle, but words w_i cannot all start with the same color in general). There is an objective that is memoryless-determined over vertex-labeled arenas [GW06] for which it is straightforward to show that it is not $\mathcal{M}_{\text{triv}}$-cycle-consistent.

The construction would however be possible for objectives admitting a so-called *neutral color*, i.e., a color that can be inserted infinitely often in every infinite word without changing its winning or losing character (for instance, 0 is neutral for parity conditions, b is neutral for $\mathsf{Reach}(a)$). In order to prove Proposition 5.5.5 with vertex-labeled arenas, v_2 could be labeled with a neutral color. Similarly, and although conditions weaker than the one for a neutral color would suffice, Lemmas 5.5.1 and 5.5.2 could also be shown with vertex-labeled arenas by labeling the "merged vertex" (called v in both cases) with this neutral color. This implies that all the results from this chapter hold using the class of vertex-labeled arenas under the existence of a neutral color. The counterexample from [GW06] indeed admits no neutral color, and edge-labeled arenas of the kind of Figure 5.5 are the typical examples in which a player needs infinite memory for this objective. The existence of a neutral color was also recently used in other works about memory requirements [CO22; Ohl23].

[GW06]: Grädel et al. (2006), *Positional Determinacy of Games with Infinitely Many Priorities*

[CO22]: Casares et al. (2022), *Characterising memory in infinite games*
[Ohl23]: Ohlmann (2023), *Characterizing Positionality in Games of Infinite Duration over Infinite Graphs*

Wrap-up of the section. Thanks to the results from this section, we deduce the first item of Theorem 5.4.1.

> **Corollary 5.5.8** (First item of Theorem 5.4.1) *If $\mathcal{M}$ suffices for both players for W in their respective countable one-player arenas, then $\sim$ has a finite index (in particular, $\mathcal{S}_W$ is well-defined and W is $\mathcal{S}_W$-prefix-independent) and W is $\mathcal{M}$-cycle-consistent.*

Proof. Follows from Propositions 5.5.3 and 5.5.5. □

In particular, we obtain from the previous result that if both players have optimal strategies based on $\mathcal{M}$ in their one-player arenas, then W is both $(\mathcal{S}_W \otimes \mathcal{M})$-prefix-independent and $(\mathcal{S}_W \otimes \mathcal{M})$-cycle-consistent (using Lemma 5.3.6). For $\mathcal{S}_W \otimes \mathcal{M}$ to be well-defined, we recall that, as described in Remark 5.3.8, we use automaton structures and memory structures as interchangeable objects in this chapter.

> **Remark 5.5.9** If we compare Example 5.3.4 ($W = \mathsf{Büchi}(a) \cap \mathsf{Büchi}(b)$) with Example 5.5.6 ($W = abC^\omega$), we see that we can easily classify the prefixes of the former, but that information is not sufficient to play optimally: we need some more information to classify cycles. For the latter, it is possible to find a memory structure classifying cycles that is

insufficient to play optimally, but a good classification of the prefixes suffices to play optimally. In general, to understand W, we need to have information about prefixes and cycles, which is why, intuitively, structure $\mathcal{S}_W \otimes \mathcal{M}$ turns out to be useful.

5.6 From properties of an objective to ω-regularity

In this section, we fix an objective $W \subseteq C^\omega$ and a memory structure $\mathcal{M} = (M, m_{\text{init}}, \alpha_{\text{upd}})$, and we assume that W **is $\mathcal{M}$-prefix-independent and $\mathcal{M}$-cycle-consistent**. Our goal is to show that W can be recognized by a parity automaton built on top of $\mathcal{M}$ and is thus ω-regular. To do that, we show in multiple steps how to assign a priority to each transition of $\mathcal{M}$ through a function $p\colon M \times C \to \{0, \ldots, n\}$ so that W is recognized by the parity automaton $(\mathcal{M}, p)$.

We write $\mathcal{M}$ throughout this section for conciseness, but when combining the results from the previous section and from this section, a typical instantiation for $\mathcal{M}$ is the product $\mathcal{S}_W \otimes \mathcal{M}'$ where $\mathcal{M}'$ is a sufficient memory structure.

5.6.1 Simplified notations

In this section, as we assume everywhere $\mathcal{M}$-prefix-independence and $\mathcal{M}$-cycle-consistency of W, we extend some notations for conciseness.

As W is $\mathcal{M}$-prefix-independent, for $m \in M$, we write $m^{-1}W$ for the set of infinite words that equals $w^{-1}W$ for any $w \in \text{col}^*(\Pi_{m_{\text{init}}, m})$. Notice in particular that $m_{\text{init}}^{-1}W = \varepsilon^{-1}W = W$. Moreover, as we consider the property of $\mathcal{M}$-cycle-consistency along with $\mathcal{M}$-prefix-independence, the definition of $\mathcal{M}$-cycle-consistency can be written by only quantifying over states of $\mathcal{M}$ and not over all finite words. The reason is that there are then only finitely many classes of finite words that matter, which correspond to the states of $\mathcal{M}$. We define a few more notations that only make sense under the $\mathcal{M}$-prefix-independent hypothesis. Let

$$\Phi_m^{\text{win}} = \{\varphi \in \Phi_m \mid (\text{col}^*(\varphi))^\omega \in m^{-1}W\}$$

be the cycles on m that induce winning words when repeated infinitely many times from m, and Φ_m^{lose} be their losing counterparts. In this case, W is $\mathcal{M}$-cycle-consistent if and only if for all $m \in M$, $(\text{col}^*(\Phi_m^{\text{win}}))^\omega \subseteq m^{-1}W$ and $(\text{col}^*(\Phi_m^{\text{lose}}))^\omega \subseteq m^{-1}\overline{W}$. We call elements of Φ_m^{win} (resp. Φ_m^{lose}) *winning* (resp. *losing*) *cycles on m*. The set of winning (resp. losing) cycles of $\mathcal{M}$ (on any state) is denoted $\Phi_{\mathcal{M}}^{\text{win}}$ (resp. $\Phi_{\mathcal{M}}^{\text{lose}}$). We write $\text{val}(\varphi)$ for the *value* of a cycle: win if $\varphi \in \Phi_{\mathcal{M}}^{\text{win}}$, and lose if $\varphi \in \Phi_{\mathcal{M}}^{\text{lose}}$.

5.6.2 Proof ideas

Our intermediate technical lemmas will focus on cycles of $\mathcal{M}$, how they relate to each other, and what happens when we combine them. Our main tool is to define a *preorder on cycles*, which will help assign priorities

to transitions of $\mathcal{M}$ — the aim being to build a DPA on top of $\mathcal{M}$ that recognizes W. Intuitively, for some state m of $\mathcal{M}$, $\varphi \in \Phi_m^{\mathsf{win}}$, and $\varphi' \in \Phi_m^{\mathsf{lose}}$, we look at which cycle "dominates" the other, that is whether the combined cycle $\varphi\varphi'$ is in Φ_m^{win} (in which case φ dominates φ') or in Φ_m^{lose} (in which case φ' dominates φ). We will formalize this and show how to extend this idea to cycles that may not share any common state.

Remark 5.6.1 One may wonder why we seek to build a DPA on top of $\mathcal{M}$ to prove that W is ω-regular, rather than a more general *deterministic Muller automaton* which would achieve the same goal. Indeed, using $\mathcal{M}$-cycle-consistency and a recent result by Casares, Colcombet, and Fijalkow [CCF21, Section 5], it is straightforward to show that we could relabel such a Muller automaton as a parity automaton on the same underlying structure recognizing the same objective.

One of the obstacles in our context is that we may start with infinitely many colors; in order to prove ω-regularity of W, we need to show at some point that many colors can be assumed to be equal (for W) in order to get finitely many classes of "equivalent" colors. The way we manage that, using the aforementioned idea of ordering cycles, actually brings us very close to directly building a relevant parity condition on top of $\mathcal{M}$ — it does not appear that our proof technique can be easily simplified by trying to obtain a Muller automaton.

Every DPA can be directly relabeled as a deterministic Muller automaton recognizing the same language, but the converse is not true in general.

[CCF21]: Casares et al. (2021), *Optimal Transformations of Games and Automata Using Muller Conditions*

Remark 5.6.2 Before tackling the proof, we also discuss the similarities between our proof and the one of Colcombet and Niwiński [CN06] about the less general prefix-independent, memoryless-determined case. Their proof compares pairs of colors directly: roughly speaking, two colors c_1 and c_2 are compared by looking at the value of cycle $c_1 c_2$.

Our first attempt was to try and use their result directly in our more general context, for instance by first taking the product of all arenas with $\mathcal{S}_W \otimes \mathcal{M}$, where $\mathcal{M}$ is a sufficient memory structure. Objective W is then memoryless-determined over this class of product arenas (from initial vertices with memory state m_{init} as a second component, see Lemma 4.2.9) and we have some kind of "prefix-independence modulo the arenas" (it does not matter how we reach a vertex in an arena).

In our case, we need to assign priorities to *transitions* of $\mathcal{S}_W \otimes \mathcal{M}$ (not just to colors). However, for a given pair of transitions, there is in general no infinite word on the alphabet of colors that just sees two specific transitions ad infinitum (it only works if the two transitions form a cycle of $\mathcal{M}$). We therefore develop a way to compare arbitrary pairs of transitions, which was not necessary in [CN06] and which is intuitively why we do not directly reuse their result or their proof.

Some of our intermediate lemmas are still very close to theirs — we then specify it explicitly — which happens when we compare cycles on the same memory state, as they can be compared directly, like colors in their context.

[CN06]: Colcombet et al. (2006), *On the positional determinacy of edge-labeled games*

This idea of "prefix-independence modulo an arena" was for instance formally defined and used in [KMST21] (it is called *tail in an arena*).

[KMST21]: Kiefer et al. (2021), *Transience in Countable MDPs*

5.6.3 Combining cycles on the same memory state

We first prove that "shifting" the start of a cycle does not alter its value.

Lemma 5.6.3 (Shift independence) *Let $m_1, m_2 \in M$ be two states of $\mathcal{M}$. Let $\pi_1 \in \Pi_{m_1,m_2}$ and $\pi_2 \in \Pi_{m_2,m_1}$; $\pi_1\pi_2$ is a cycle on m_1 and $\pi_2\pi_1$ is a cycle on m_2. Then, $\mathsf{val}(\pi_1\pi_2) = \mathsf{val}(\pi_2\pi_1)$.*

Proof. For all $w_1 \in \mathsf{col}^*(\Pi_{m_1,m_2})$ and $w_2 \in C^\omega$, notice that

$$
\begin{aligned}
w_1 w_2 \in m_1^{-1}W &\iff \exists w \in \mathsf{col}^*(\Pi_{m_{\mathsf{init}},m_1}), w_1 w_2 \in w^{-1}W \\
&\iff \exists w \in \mathsf{col}^*(\Pi_{m_{\mathsf{init}},m_1}), w_2 \in (ww_1)^{-1}W \\
&\iff \exists w' \in \mathsf{col}^*(\Pi_{m_{\mathsf{init}},m_2}), w_2 \in (w')^{-1}W \\
&\iff w_2 \in m_2^{-1}W.
\end{aligned}
$$

The third equivalence is due to the fact that ww_1 is in $\mathsf{col}^*(\Pi_{m_{\mathsf{init}},m_2})$ for the left-to-right implication, and to $\mathcal{M}$-prefix-independence for the right-to-left implication; if there exists $w' \in \mathsf{col}^*(\Pi_{m_{\mathsf{init}},m_2})$ such that $w_2 \in (w')^{-1}W$, then the same is true for any word in $\mathsf{col}^*(\Pi_{m_{\mathsf{init}},m_2})$.

Going back to the statement of the lemma, we have that

$$
\begin{aligned}
\pi_1\pi_2 \in \Phi_{m_1}^{\mathsf{win}} &\iff (\mathsf{col}^*(\pi_1\pi_2))^\omega \in m_1^{-1}W \\
&\iff \mathsf{col}^*(\pi_1)(\mathsf{col}^*(\pi_2\pi_1))^\omega \in m_1^{-1}W \\
&\qquad \text{as } (\mathsf{col}^*(\pi_1\pi_2))^\omega = \mathsf{col}^*(\pi_1)(\mathsf{col}^*(\pi_2\pi_1))^\omega \\
&\iff (\mathsf{col}^*(\pi_2\pi_1))^\omega \in m_2^{-1}W \\
&\qquad \text{by the above property as } \mathsf{col}^*(\pi_1) \in \mathsf{col}^*(\Pi_{m_1,m_2}) \\
&\iff \pi_2\pi_1 \in \Phi_{m_2}^{\mathsf{win}}.
\end{aligned}
$$

Hence, the values of $\pi_1\pi_2$ and $\pi_2\pi_1$ always coincide. □

In particular, this result implies that swapping two cycles on the same memory state does not alter the value: if $\varphi, \varphi' \in \Phi_m$, then $\mathsf{val}(\varphi\varphi') = \mathsf{val}(\varphi'\varphi)$.

The next two lemmas are used to show that although cycles of $\mathcal{M}$ that are taken infinitely often might have an impact on the result of a play, their *relative number of repetitions* is not relevant (i.e., $\mathsf{val}(\varphi\varphi') = \mathsf{val}(\varphi^k(\varphi')^l)$ for any $k, l \geq 1$). These two proofs and statements are very close to [CN06, Lemmas 9, 10, and 11] and are a direct generalization to a larger class of objectives.

[CN06]: Colcombet et al. (2006), *On the positional determinacy of edge-labeled games*

Lemma 5.6.4 *Let $m \in M$. Let $\Lambda, \Lambda' \subseteq \Phi_m$ be non-empty sets of cycles on m. We have*

$$
\forall \varphi' \in \Lambda', \exists \varphi \in \Lambda, \varphi\varphi' \in \Phi_m^{\mathsf{win}} \implies \exists \varphi \in \Lambda, \forall \varphi' \in \Lambda', \varphi\varphi' \in \Phi_m^{\mathsf{win}}.
$$

This lemma says that if all cycles from Λ' can be made winning by adjoining them a cycle from Λ, then we can actually find a single cycle from Λ that makes all cycles from Λ' winning.

Proof. We assume the premise of the implication, and by contradiction, we assume that the conclusion is false. We therefore assume that

$$\forall \varphi' \in \Lambda', \exists \varphi \in \Lambda, \varphi \varphi' \in \Phi_m^{\mathsf{win}} \quad \text{and} \quad \forall \varphi \in \Lambda, \exists \varphi' \in \Lambda', \varphi \varphi' \in \Phi_m^{\mathsf{lose}}.$$

Let φ_1 be any word in Λ. We build inductively an infinite sequence starting with φ_1 by alternating the use of the two assumptions. For $i \geq 1$, we take $\varphi_i' \in \Lambda'$ such that $\varphi_i \varphi_i' \in \Phi_m^{\mathsf{lose}}$ (using the second assumption), and we then take $\varphi_{i+1} \in \Lambda$ such that $\varphi_{i+1} \varphi_i' \in \Phi_m^{\mathsf{win}}$ (using the first assumption).

We consider the infinite sequence $\varphi_1 \varphi_1' \varphi_2 \varphi_2' \varphi_3 \ldots \in (M \times C)^\omega$ such that for all $i \geq 1$, $\varphi_i \varphi_i' \in \Phi_m^{\mathsf{lose}}$ and $\varphi_i' \varphi_{i+1} \in \Phi_m^{\mathsf{win}}$ (we use that the order of cycles on m does not matter, shown in Lemma 5.6.3). We show that the infinite word $\mathsf{col}^\omega(\varphi_1 \varphi_1' \varphi_2 \varphi_2' \ldots)$ is both in $m^{-1}W$ and in $m^{-1}\overline{W}$ by pairing cycles in two different ways:

- the infinite sequence $(\varphi_1 \varphi_1')(\varphi_2 \varphi_2') \ldots$ is a sequence of losing cycles on m and its projection to colors is therefore in $m^{-1}\overline{W}$ by using $\mathcal{M}$-cycle-consistency.
- the infinite word $\mathsf{col}^\omega(\varphi_1(\varphi_1' \varphi_2)(\varphi_2' \varphi_3) \ldots)$ is in $m^{-1}W$ if and only if $\mathsf{col}^\omega((\varphi_1' \varphi_2)(\varphi_2' \varphi_3) \ldots)$ is in $m^{-1}W$ by using that $\varphi_1 \in \Phi_m$ and $\mathcal{M}$-prefix-independence of W. The sequence $(\varphi_1' \varphi_2)(\varphi_2' \varphi_3) \ldots$ is a sequence of winning cycles on m and its projection to colors is in $m^{-1}W$ by using $\mathcal{M}$-cycle-consistency.

As $m^{-1}W \cap m^{-1}\overline{W} = \emptyset$, we have our contradiction. □

Lemma 5.6.5 (Repetition independence) *Let $m \in M$. Let $\varphi, \varphi' \in \Phi_m$ such that $\varphi \varphi' \in \Phi_m^{\mathsf{win}}$. We have $\varphi(\varphi')^+ \subseteq \Phi_m^{\mathsf{win}}$.*

Proof. We have that φ or φ' is in Φ_m^{win} — otherwise, $\varphi \varphi'$ would be in Φ_m^{lose} by $\mathcal{M}$-cycle-consistency. If φ' is in Φ_m^{win}, we notice that any element of $\varphi(\varphi')^+$ can be written as $(\varphi \varphi')(\varphi')^n$ for some $n \geq 0$, which is a combination of winning cycles on m. Using $\mathcal{M}$-cycle-consistency, we thus get that $\varphi(\varphi')^+ \subseteq \Phi_m^{\mathsf{win}}$.

It is left to deal with the case $\varphi \in \Phi_m^{\mathsf{win}}$ and $\varphi' \in \Phi_m^{\mathsf{lose}}$. We first show by induction that for $n \geq 1$, $\varphi^n(\varphi')^n \in \Phi_m^{\mathsf{win}}$. This is true by hypothesis for $n = 1$. We now assume it is true for some $n \geq 1$, and we show it is true for $n + 1$. By Lemma 5.6.3, we have that $\varphi^{n+1}(\varphi')^{n+1} \in \Phi_m^{\mathsf{win}}$ if and only if $\varphi^{n+1}(\varphi')^{n+1}\varphi = (\varphi^n(\varphi')^n)(\varphi' \varphi) \in \Phi_m^{\mathsf{win}}$, by swapping the order of φ and $\varphi^n(\varphi')^{n+1}$. By induction hypothesis, $\varphi^n(\varphi')^n \in \Phi_m^{\mathsf{win}}$; by hypothesis and by Lemma 5.6.3, $\varphi' \varphi \in \Phi_m^{\mathsf{win}}$. Therefore, by $\mathcal{M}$-cycle-consistency, $(\varphi^n(\varphi')^n)(\varphi' \varphi)$ is also in Φ_m^{win}.

We now define $\Lambda = \varphi^+$ and $\Lambda' = (\varphi')^+$. We have that for all elements $(\varphi')^n$ of Λ' (with $n \geq 1$), we have that φ^n (an element of Λ) is such that $\varphi^n(\varphi')^n \in \Phi_m^{\mathsf{win}}$. Therefore the hypothesis of Lemma 5.6.4 is satisfied for Λ and Λ', which implies that there exists $n \geq 1$ such that $\varphi^n(\varphi')^+ \subseteq \Phi_m^{\mathsf{win}}$.

We assume w.l.o.g. that $n = \min\{n \in \mathbb{N} \mid \varphi^n(\varphi')^+ \subseteq \Phi_m^{\mathsf{win}}\}$. For all $k \in \mathbb{N}$ such that $k \geq n$, we also have that $\varphi^k(\varphi')^+ = \varphi^{k-n}(\varphi^n(\varphi')^+) \subseteq \Phi_m^{\mathsf{win}}$ by $\mathcal{M}$-cycle-consistency. We intend to show that $n = 1$, which would end the proof of the lemma as this would show that $\varphi^1(\varphi')^+ = \varphi(\varphi')^+ \subseteq \Phi_m^{\mathsf{win}}$.

We assume by contradiction that $n > 1$. Then there must exist $k \in \mathbb{N}$ such that $\varphi^{n-1}(\varphi')^k \in \Phi_m^{\mathsf{lose}}$. We also have that $(\varphi')^k \varphi^{n-1}$ is in Φ_m^{lose} by Lemma 5.6.3, which implies that $\varphi^{n-1}(\varphi')^k(\varphi')^k \varphi^{n-1}$ is also in Φ_m^{lose} by $\mathcal{M}$-cycle-consistency. But then by Lemma 5.6.3, this cycle has the same value as $\varphi^{2n-2}(\varphi')^{2k}$, which must therefore be in Φ_m^{lose}. This is a contradiction since $n > 1$ implies that $2n - 2 \geq n$.

We conclude that $\varphi(\varphi')^+ \subseteq \Phi_m^{\mathsf{win}}$. $\qquad\square$

Thanks to this result, we can now show that any two consecutive cycles on the same memory state can always be swapped without altering the value of a longer cycle.

> **Corollary 5.6.6** (Cycle-order independence) *Let $m \in M$. Let $\varphi_1, \varphi_2, \varphi_3 \in \Phi_m$. Then,* $\mathsf{val}(\varphi_1\varphi_2\varphi_3) = \mathsf{val}(\varphi_1\varphi_3\varphi_2)$.

Proof. We assume by contradiction that cycles $\varphi_1\varphi_2\varphi_3$ and $\varphi_1\varphi_3\varphi_2$ have a different value; w.l.o.g., that $\varphi_1\varphi_2\varphi_3 \in \Phi_m^{\mathsf{win}}$ and that $\varphi_1\varphi_3\varphi_2 \in \Phi_m^{\mathsf{lose}}$. By $\mathcal{M}$-cycle-consistency, at least one cycle among φ_1, φ_2 and φ_3 is winning and one is losing. We assume w.l.o.g. that $\varphi_1 \in \Phi_m^{\mathsf{win}}$ and $\varphi_2 \in \Phi_m^{\mathsf{lose}}$. We also assume that $\varphi_3 \in \Phi_m^{\mathsf{win}}$; the other case can be dealt with by symmetry. Notice that we necessarily have that $\varphi_3\varphi_2$ is in Φ_m^{lose}; otherwise, $\varphi_1\varphi_3\varphi_2 = \varphi_1(\varphi_3\varphi_2)$ would be in Φ_m^{win} by $\mathcal{M}$-cycle-consistency. For the same reason, $\varphi_2\varphi_1$ is in Φ_m^{lose}. We have

$$
\begin{aligned}
\mathsf{win} = \mathsf{val}(\varphi_1\varphi_2\varphi_3) && \text{by hypothesis} \\
= \mathsf{val}((\varphi_3\varphi_1)\varphi_2) && \text{by Lemma 5.6.3} \\
= \mathsf{val}((\varphi_3\varphi_1)(\varphi_2)^2) && \text{by Lemma 5.6.5} \\
= \mathsf{val}(\varphi_2\varphi_3\varphi_1\varphi_2) && \text{by Lemma 5.6.3.}
\end{aligned}
$$

However, this last cycle can be written as a combination of two losing cycles $(\varphi_2\varphi_3)$ and $(\varphi_1\varphi_2)$, and should therefore be losing by $\mathcal{M}$-cycle-consistency. This is a contradiction. $\qquad\square$
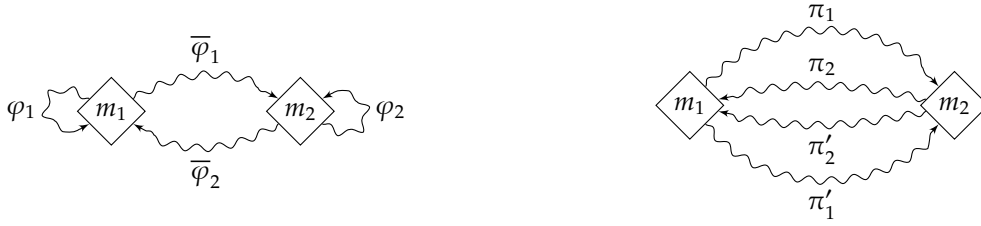
Figure 5.7: Depiction of the statement of Corollary 5.6.7 (left) and Lemma 5.6.8 (right).

5.6.4 Combining cycles on different memory states

We can now strengthen Lemma 5.6.5 (dubbed "repetition independence") to show that even "non-consecutive subcycles" in a longer cycle can be repeated without altering the value of the long cycle.

> **Corollary 5.6.7** (Repetition independence, strong version) *Let $m_1, m_2 \in$ M. Let $\varphi_1 \in \Phi_{m_1}$, $\varphi_2 \in \Phi_{m_2}$, $\overline{\varphi}_1 \in \Pi_{m_1,m_2}$, and $\overline{\varphi}_2 \in \Pi_{m_2,m_1}$. Then, $\mathsf{val}(\varphi_1 \overline{\varphi}_1 \varphi_2 \overline{\varphi}_2) = \mathsf{val}(\varphi_1 (\overline{\varphi}_1 \overline{\varphi}_2)^n \overline{\varphi}_1 \varphi_2 \overline{\varphi}_2)$ for all $n \geq 0$.*

The situation is depicted in Figure 5.7 (left). Notice first that we can see $\varphi_1 \overline{\varphi}_1 \varphi_2 \overline{\varphi}_2$ as a combination of two cycles φ_1 and $\overline{\varphi}_1 \varphi_2 \overline{\varphi}_2$ on m_1, we therefore already know that the value of $\varphi_1 \overline{\varphi}_1 \varphi_2 \overline{\varphi}_2$ on m_1 is the same as the one of $(\varphi_1)^k (\overline{\varphi}_1 \varphi_2 \overline{\varphi}_2)^l$ for all $k, l \geq 1$. This second cycle can be seen as two cycles $\overline{\varphi}_2 \overline{\varphi}_1$ and φ_2 on m_2, we therefore know that the value of $\varphi_2 \overline{\varphi}_2 \overline{\varphi}_1$ on m_2 is the same as the one of $(\varphi_2)^k (\overline{\varphi}_2 \overline{\varphi}_1)^l$ for all $k, l \geq 1$. However, these two facts do not directly give the result as cycle $\overline{\varphi}_1 \overline{\varphi}_2$ does not appear "consecutively" in $\varphi_1 \overline{\varphi}_1 \varphi_2 \overline{\varphi}_2$.

Proof of Corollary 5.6.7. We have that

$$
\begin{aligned}
\mathsf{val}(\varphi_1 \overline{\varphi}_1 \varphi_2 \overline{\varphi}_2) &= \mathsf{val}(\varphi_1 (\overline{\varphi}_1 \varphi_2 \overline{\varphi}_2)(\overline{\varphi}_1 \varphi_2 \overline{\varphi}_2)) && \text{by Lemma 5.6.5 on } m_1 \\
&= \mathsf{val}(\varphi_1 \overline{\varphi}_1 (\varphi_2)(\overline{\varphi}_2 \overline{\varphi}_1) \varphi_2 \overline{\varphi}_2) \\
&= \mathsf{val}(\varphi_1 \overline{\varphi}_1 (\overline{\varphi}_2 \overline{\varphi}_1)(\varphi_2) \varphi_2 \overline{\varphi}_2) && \text{by Lemma 5.6.6 on } m_2 \\
&= \mathsf{val}(\varphi_1 (\overline{\varphi}_1 \overline{\varphi}_2) \overline{\varphi}_1 (\varphi_2)^2 \overline{\varphi}_2) \\
&= \mathsf{val}(\varphi_1 (\overline{\varphi}_1 \overline{\varphi}_2) \overline{\varphi}_1 \varphi_2 \overline{\varphi}_2) && \text{by Lemma 5.6.5 on } m_2.
\end{aligned}
$$

This shows the result for $n = 1$; applying Lemma 5.6.5 gives the result for all $n \geq 1$. □

Another important property that will help define an interesting preorder on cycles is that the value of a combination of two cycles is independent of the memory state chosen to compare pairs of cycles: if two cycles both go through two states m_1 and m_2 of $\mathcal{M}$, then combining them around m_1 or around m_2 yields the same value.

Lemma 5.6.8 (Crossing-point independence) *Let $m_1, m_2 \in M$ be two states of $\mathcal{M}$. Let $\pi_1, \pi'_1 \in \Pi_{m_1, m_2}$ and $\pi_2, \pi'_2 \in \Pi_{m_2, m_1}$. We have that* $\mathsf{val}(\pi_1 \pi_2 \pi'_1 \pi'_2) = \mathsf{val}(\pi_2 \pi_1 \pi'_2 \pi'_1)$.

The intuition of this lemma is that if we take two cycles (in the statement, $\pi_1 \pi_2$ and $\pi'_1 \pi'_2$) that have two common states (m_1 and m_2), the chosen starting state to combine the two cycles (the combination is $(\pi_1 \pi_2)(\pi'_1 \pi'_2)$ if m_1 is chosen, and $(\pi_2 \pi_1)(\pi'_2 \pi'_1)$ if m_2 is chosen) has no impact on the value of the combination. This situation is depicted in Figure 5.7 (right).

Proof of Lemma 5.6.8. If $\pi_1 \pi_2$ and $\pi'_1 \pi'_2$ are both in $\Phi^{\mathsf{win}}_{m_1}$ or both in $\Phi^{\mathsf{lose}}_{m_1}$, then $\pi_2 \pi_1$ and $\pi'_2 \pi'_1$ are also respectively both in $\Phi^{\mathsf{win}}_{m_2}$ or both in $\Phi^{\mathsf{lose}}_{m_2}$ by Lemma 5.6.3. Therefore, we have our result using $\mathcal{M}$-cycle-consistency.

For the remaining cases, we assume w.l.o.g. that $\pi_1 \pi_2 \in \Phi^{\mathsf{win}}_{m_1}$ and $\pi'_1 \pi'_2 \in \Phi^{\mathsf{lose}}_{m_2}$. We will assume (again w.l.o.g.) that combining them is winning, i.e., that $\pi_1 \pi_2 \pi'_1 \pi'_2 \in \Phi^{\mathsf{win}}_{m_1}$. Our goal is to show that $\pi_2 \pi_1 \pi'_2 \pi'_1$ is also in $\Phi^{\mathsf{win}}_{m_2}$. We assume by contradiction that it is not, i.e., that $\pi_2 \pi_1 \pi'_2 \pi'_1 \in \Phi^{\mathsf{lose}}_{m_2}$.

Observe that as $\pi_1 \pi_2 \pi'_1 \pi'_2 \in \Phi^{\mathsf{win}}_{m_1}$, we also have $(\pi_2 \pi'_1)(\pi'_2 \pi_1) \in \Phi^{\mathsf{win}}_{m_2}$ by Lemma 5.6.3. Hence, at least one of $\pi_2 \pi'_1$ and $\pi'_2 \pi_1$ must be a winning cycle on m_2, otherwise their combination would be losing on m_2 by $\mathcal{M}$-cycle-consistency. Equivalently, by Lemma 5.6.3, at least one of $\pi'_1 \pi_2$ and $\pi_1 \pi'_2$ must be a winning cycle on m_1.

Similarly, as $\pi_2 \pi_1 \pi'_2 \pi'_1$ is in $\Phi^{\mathsf{lose}}_{m_2}$, we have that $(\pi_1 \pi'_2)(\pi'_1 \pi_2)$ is in $\Phi^{\mathsf{lose}}_{m_1}$. Hence, at least one of $\pi_1 \pi'_2$ and $\pi'_1 \pi_2$ is a losing cycle on m_1 by $\mathcal{M}$-cycle-consistency.

Our conclusions imply that exactly one of $\pi_1 \pi'_2$ and $\pi'_1 \pi_2$ is winning on m_1, and exactly one is losing on m_1. Without loss of generality, we assume that $\pi_1 \pi'_2 \in \Phi^{\mathsf{win}}_{m_1}$ and $\pi'_1 \pi_2 \in \Phi^{\mathsf{lose}}_{m_1}$.

We now have a value for all four two-word cycles on m_1 (and therefore for all four two-word cycles on m_2 by Lemma 5.6.3): $\pi_1 \pi_2$ and $\pi_1 \pi'_2$ are in $\Phi^{\mathsf{win}}_{m_1}$, and $\pi'_1 \pi_2$ and $\pi'_1 \pi'_2$ are in $\Phi^{\mathsf{lose}}_{m_1}$. If we look at four-word cycles, we have already assumed w.l.o.g. that $\pi_1 \pi_2 \pi'_1 \pi'_2 \in \Phi^{\mathsf{win}}_{m_1}$ and that $\pi_1 \pi'_2 \pi'_1 \pi_2 \in \Phi^{\mathsf{lose}}_{m_1}$. We still do not know whether $\pi_1 \pi_2 \pi'_1 \pi_2$ and $\pi_1 \pi'_2 \pi'_1 \pi'_2$ are winning or losing — no matter how we express them as two two-word cycles, one two-word cycle is winning and the other one is losing. We study the value of these two four-word cycles.

Consider the cycle $(\pi_2 \pi'_1 \pi'_2 \pi_1)(\pi'_2 \pi_1 \pi_2 \pi'_1)$ on m_2. It is winning, since $\pi_2 \pi'_1 \pi'_2 \pi_1$ and $\pi'_2 \pi_1 \pi_2 \pi'_1$ are both in $\Phi^{\mathsf{win}}_{m_2}$: this can be shown using Lemma 5.6.3 and the fact that $\pi_1 \pi_2 \pi'_1 \pi'_2$ is in $\Phi^{\mathsf{win}}_{m_1}$. Therefore, by shifting the start of the cycle, $(\pi'_1 \pi'_2 \pi_1 \pi'_2)(\pi_1 \pi_2 \pi'_1 \pi_2)$ is in $\Phi^{\mathsf{win}}_{m_1}$. By $\mathcal{M}$-cycle-consistency, this means that at least one of $\pi'_1 \pi'_2 \pi_1 \pi'_2$ (equivalently, $\pi_1 \pi'_2 \pi'_1 \pi'_2$) and $\pi_1 \pi_2 \pi'_1 \pi_2$ is winning on m_1.

Similarly, we have that the cycle $(\pi_2\pi_1\pi_2'\pi_1')(\pi_2'\pi_1'\pi_2\pi_1)$ is in $\Phi_{m_2}^{\text{lose}}$. Therefore, by shifting the start of the cycle, $(\pi_1\pi_2'\pi_1'\pi_2')(\pi_1'\pi_2\pi_1\pi_2)$ is in $\Phi_{m_1}^{\text{lose}}$. This means that at least one of $\pi_1\pi_2'\pi_1'\pi_2'$ and $\pi_1'\pi_2\pi_1\pi_2$ (equivalently, $\pi_1\pi_2\pi_1'\pi_2$) is in $\Phi_{m_1}^{\text{lose}}$.

Our conclusions imply that exactly one of $\pi_1\pi_2\pi_1'\pi_2$ and $\pi_1\pi_2'\pi_1'\pi_2'$ is winning on m_1, and one is losing on m_1. We consider both cases and draw a contradiction in each case.

Assume that $\text{val}(\pi_1\pi_2\pi_1'\pi_2) = \text{lose}$. Now consider the cycle

$$\psi = (\pi_2'\pi_1)(\pi_2\pi_1')^2$$

on m_2. We have

$$
\begin{aligned}
\text{val}(\psi) &= \text{val}(\pi_2'\pi_1\pi_2\pi_1') && \text{by Lemma 5.6.5} \\
&= \text{val}(\pi_1\pi_2\pi_1'\pi_2') && \text{by Lemma 5.6.3} \\
&= \text{win} && \text{by hypothesis.}
\end{aligned}
$$

However, we also have

$$\text{val}(\psi) = \text{val}((\pi_1\pi_2\pi_1'\pi_2)(\pi_1'\pi_2')) \qquad \text{by Lemma 5.6.3,}$$

and since $\text{val}(\pi_1\pi_2\pi_1'\pi_2) = \text{val}(\pi_1'\pi_2') = \text{lose}$, we also have $\text{val}(\psi) = \text{lose}$ by $\mathcal{M}$-cycle-consistency. This is a contradiction.

Assume now that $\text{val}(\pi_1\pi_2'\pi_1'\pi_2') = \text{lose}$. Now consider the cycle

$$\psi = (\pi_1\pi_2)(\pi_1'\pi_2')^2$$

on m_1. We have

$$
\begin{aligned}
\text{val}(\psi) &= \text{val}(\pi_1\pi_2\pi_1'\pi_2') && \text{by Lemma 5.6.5} \\
&= \text{win} && \text{by hypothesis.}
\end{aligned}
$$

However, we also have

$$\text{val}(\psi) = \text{val}((\pi_2\pi_1')(\pi_2'\pi_1'\pi_2'\pi_1)) \qquad \text{by Lemma 5.6.3,}$$

and since $\text{val}(\pi_1'\pi_2) = \text{val}(\pi_1\pi_2'\pi_1'\pi_2') = \text{lose}$, we also have $\text{val}(\psi) = \text{lose}$ by $\mathcal{M}$-cycle-consistency. This is a contradiction. $\qquad\square$

Remark 5.6.9 A consequence of the previous lemma is that when two cycles $\varphi, \varphi' \in \Phi_{\mathcal{M}}$ share at least one common state (i.e., $\text{st}(\varphi) \cap \text{st}(\varphi') \neq \emptyset$), we can write $\varphi\varphi'$ for any cycle that, starting from a common state, sees first φ and then φ', without necessarily specifying on which common state the cycle starts; we allow such a shortcut as the value of $\varphi\varphi'$ is not impacted by the choice of the common memory state. This convention is used in the following definition.

5.6.5 Competing cycles

We would now like to define a way to compare two cycles that may not share a common state.

> **Definition 5.6.10** *Let* $\varphi, \varphi' \in \Phi_{\mathcal{M}}$ *with* $\mathsf{val}(\varphi) \neq \mathsf{val}(\varphi')$. *We say that* φ *and* φ' *are competing if there exists* $\overline{\varphi} \in \Phi_{\mathcal{M}}$ *such that* $\mathsf{st}(\varphi) \cap \mathsf{st}(\overline{\varphi}) \neq \emptyset$, $\mathsf{st}(\varphi') \cap \mathsf{st}(\overline{\varphi}) \neq \emptyset$, $\mathsf{val}(\varphi\overline{\varphi}) = \mathsf{val}(\varphi)$, *and* $\mathsf{val}(\varphi'\overline{\varphi}) = \mathsf{val}(\varphi')$. *In this case, we say that* $\overline{\varphi}$ *is a* witness *that* φ *and* φ' *are competing, or that* the competition of φ *and* φ' *is witnessed by* $\overline{\varphi}$.

In what follows, cycles with a line above them denote witnesses.

Our requirement for cycle $\overline{\varphi}$ means that it intersects the states of both φ and φ', but is not influential enough to "alter" the values of φ and φ' when it is combined with them. If $\mathsf{val}(\varphi) \neq \mathsf{val}(\varphi')$ and $\mathsf{st}(\varphi) \cap \mathsf{st}(\varphi') \neq \emptyset$, then if $\mathsf{val}(\varphi\varphi') = \mathsf{val}(\varphi)$ (resp. $\mathsf{val}(\varphi\varphi') = \mathsf{val}(\varphi')$), we have that φ' (resp. φ) witnesses that φ and φ' are competing (the argument uses Lemma 5.6.5). In short, any two cycles of opposite values that share a common state are competing, and two cycles of opposite values that do not share a common state may or may not be competing.

If two cycles are competing, we want to determine which one *dominates* the other.

> **Definition 5.6.11** *Let* $\varphi, \varphi' \in \Phi_{\mathcal{M}}$ *with* $\mathsf{val}(\varphi) \neq \mathsf{val}(\varphi')$ *be two competing cycles, and* $\overline{\varphi}$ *be a witness of this competition. For some* $m \in \mathsf{st}(\varphi)$ *and* $m' \in \mathsf{st}(\varphi')$, *it is thus possible to decompose* $\overline{\varphi}$ *as two (possibly empty) paths* $\overline{\varphi}_1$ *and* $\overline{\varphi}_2$ *such that* $\overline{\varphi} = \overline{\varphi}_1\overline{\varphi}_2$, $\overline{\varphi}_1 \in \Pi_{m,m'}$, *and* $\overline{\varphi}_2 \in \Pi_{m',m}$. *We define that* φ dominates φ' *if* $\mathsf{val}(\varphi\overline{\varphi}_1\varphi'\overline{\varphi}_2) = \mathsf{val}(\varphi)$, *and* φ' dominates φ *if* $\mathsf{val}(\varphi\overline{\varphi}_1\varphi'\overline{\varphi}_2) = \mathsf{val}(\varphi')$.

To be well-defined, this *domination* notion needs to be independent of the choice of witness.

> **Lemma 5.6.12** (Witness independence) *Let* $\varphi, \varphi' \in \Phi_{\mathcal{M}}$ *with* $\mathsf{val}(\varphi) \neq \mathsf{val}(\varphi')$. *Let* $\overline{\varphi}_1, \overline{\varphi}_2 \in \Phi_{\mathcal{M}}$ *be two witnesses that* φ *and* φ' *are competing. Then,* φ *dominates* φ' *taking* $\overline{\varphi}_1$ *as a witness if and only if* φ *dominates* φ' *taking* $\overline{\varphi}_2$ *as a witness.*

Proof. We assume w.l.o.g. that $\mathsf{val}(\varphi) = \mathsf{win}$ and $\mathsf{val}(\varphi') = \mathsf{lose}$. As $\overline{\varphi}_1$ witnesses that φ and φ' are competing, there exists $m_1 \in \mathsf{st}(\varphi)$, $m_1' \in \mathsf{st}(\varphi')$ such that $\overline{\varphi}_1 = \overline{\varphi}_{1,1}\overline{\varphi}_{1,2}$ with $\overline{\varphi}_{1,1} \in \Pi_{m_1,m_1'}$, $\overline{\varphi}_{1,2} \in \Pi_{m_1',m_1}$. Similarly, as $\overline{\varphi}_2$ witnesses that φ and φ' are competing, there exists $m_2 \in \mathsf{st}(\varphi)$, $m_2' \in \mathsf{st}(\varphi')$ such that $\overline{\varphi}_2 = \overline{\varphi}_{2,1}\overline{\varphi}_{2,2}$ with $\overline{\varphi}_{2,1} \in \Pi_{m_2,m_2'}$, $\overline{\varphi}_{2,2} \in \Pi_{m_2',m_2}$. We can also write $\varphi = \varphi_1\varphi_2$ with $\varphi_1 \in \Pi_{m_1,m_2}$ and $\varphi_2 \in \Pi_{m_2,m_1}$, and $\varphi' = \varphi_1'\varphi_2'$ with $\varphi_1 \in \Pi_{m_1',m_2'}$ and $\varphi_2' \in \Pi_{m_2',m_1'}$.

The situation is depicted in Figure 5.8. Note that it is possible that $m_1 = m_2$ (in which case we can assume $\varphi = \varphi_1$, $\varphi_2 = (m_1, \_)$) or similarly that
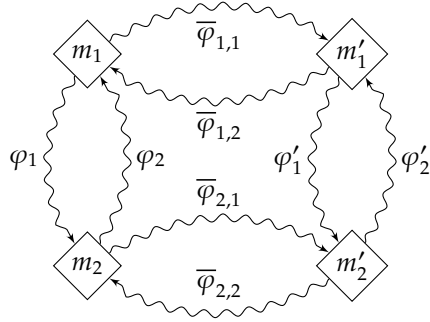
Figure 5.8: Situation in the proof of Lemma 5.6.12.

$m_1 = m_1'$, $m_1' = m_2'$, and/or $m_2 = m_2'$.

We assume by contradiction that φ dominates φ' taking $\overline{\varphi}_1$ as a witness, but that φ' dominates φ taking $\overline{\varphi}_2$ as a witness. In other words, $\mathsf{val}(\varphi_1 \varphi_2 \overline{\varphi}_{1,1} \varphi_1' \varphi_2' \overline{\varphi}_{1,2}) = \mathsf{win}$ and $\mathsf{val}(\varphi_2 \varphi_1 \overline{\varphi}_{2,1} \varphi_2' \varphi_1' \overline{\varphi}_{2,2}) = \mathsf{lose}$. We consider the concatenation of both these cycles, shifting the second one to make it a cycle on m_1,

$$\psi = (\varphi_1 \varphi_2 \overline{\varphi}_{1,1} \varphi_1' \varphi_2' \overline{\varphi}_{1,2})(\varphi_1 \overline{\varphi}_{2,1} \varphi_2' \varphi_1' \overline{\varphi}_{2,2} \varphi_2).$$

It is possible to express ψ directly as a combination of two losing cycles: $\overline{\varphi}_{1,1} \varphi_1' \varphi_2' \overline{\varphi}_{1,2}$ is losing (by definition of witness), and

$\mathsf{val}(\varphi_1 \overline{\varphi}_{2,1} \varphi_2' \varphi_1' \overline{\varphi}_{2,2} \varphi_2 \varphi_1 \varphi_2)$

$\quad = \mathsf{val}(\overline{\varphi}_{2,1} \varphi_2' \varphi_1' \overline{\varphi}_{2,2} (\varphi_2 \varphi_1)^2)$ by Lemma 5.6.3

$\quad = \mathsf{val}(\overline{\varphi}_{2,1} \varphi_2' \varphi_1' \overline{\varphi}_{2,2} \varphi_2 \varphi_1)$ by Lemma 5.6.5

$\quad = \mathsf{lose}$ as φ' dominates φ with witness $\overline{\varphi}_2$.

Cycle ψ is therefore losing by $\mathcal{M}$-cycle-consistency.

Now, notice that ψ can be written as three cycles on m_2' after being shifted in the following way:

$$\mathsf{val}(\psi) = \mathsf{val}((\varphi_2' \overline{\varphi}_{1,2} \varphi_1 \overline{\varphi}_{2,1})(\varphi_2' \varphi_1')(\overline{\varphi}_{2,2} \varphi_2 \varphi_1 \varphi_2 \overline{\varphi}_{1,1} \varphi_1')).$$

By Lemma 5.6.6 ("cycle-order independence"), it has the same value as

$$\mathsf{val}(\psi) = \mathsf{val}((\varphi_2' \overline{\varphi}_{1,2} \varphi_1 \overline{\varphi}_{2,1})(\overline{\varphi}_{2,2} \varphi_2 \varphi_1 \varphi_2 \overline{\varphi}_{1,1} \varphi_1')(\varphi_2' \varphi_1')).$$

As before, this cycle can be shifted and written as two winning cycles $\varphi_1 \overline{\varphi}_{2,1} \overline{\varphi}_{2,2} \varphi_2$ and $\varphi_1 \varphi_2 \overline{\varphi}_{1,1} (\varphi_1' \varphi_2')^2 \overline{\varphi}_{1,2}$, and is therefore winning by $\mathcal{M}$-cycle-consistency.

Cycle ψ is both winning and losing, a contradiction. $\qquad\qquad\square$

Example 5.6.13 We illustrate competition and domination of cycles on a parity automaton (even though at this point in the proof, we have not yet shown that W is ω-regular). We consider the parity automa-

Figure 5.9: A parity automaton $\mathcal{P}$ (left) with $C = \{a, b, c\}$ and underlying structure $\mathcal{M}$ used in Example 5.6.13. A diagram (right) showing the relations between the elements of $\Phi_{\mathcal{M}}/_{\simeq}$ ordered by partial preorder $\lhd$, discussed in Example 5.6.16.

ton $\mathcal{P}$ from Figure 5.9, denoting by $\mathcal{M}$ its underlying structure, with objective $W = \mathcal{L}(\mathcal{P})$ (left). Objective W is $\mathcal{M}$-prefix-independent and $\mathcal{M}$-cycle-consistent (Lemma 5.3.7). We give a few examples of competition and domination between cycles. The winning cycle $(m_1, a)(m_2, a)$ dominates losing cycle (m_1, b) but is dominated by losing cycle (m_1, c). Cycle (m_2, b) is winning but is not competing with (m_1, b). In particular, their competition is not witnessed by $(m_1, a)(m_2, a)$ since combining it with (m_1, b) alters its value $((m_1, b)$ is losing but $(m_1, b)(m_1, a)(m_2, a)$ is winning) — another potential witness is $(m_1, c)(m_1, a)(m_2, a)$, but combining it with (m_2, b) alters its value. However, cycle (m_2, b) is competing with and dominated by (m_1, c): their competition is witnessed, e.g., by $(m_1, a)(m_2, a)$ and by $(m_1, b)(m_1, a)(m_2, a)$.

> We recall that we use the "lim sup" variant of the parity acceptance condition (what matters is the *maximal* priority seen infinitely often).

5.6.6 Preorder on cycles

For a winning (resp. losing) cycle $\varphi \in \Phi_{\mathcal{M}}$, we define $\mathsf{comp}(\varphi)$ as the set of losing (resp. winning) cycles that φ is competing with, and $\mathsf{domBy}(\varphi)$ as the set of losing (resp. winning) cycles from $\mathsf{comp}(\varphi)$ that are dominated by φ.

We define an ordering $\lhd$ of cycles based on these notions. We write $\varphi' \lhd \varphi$ if $\varphi' \in \mathsf{domBy}(\varphi)$. We extend this definition to cycles with the same value: if $\mathsf{val}(\varphi_1) = \mathsf{val}(\varphi_2)$ and there exists a cycle φ' such that $\mathsf{val}(\varphi') \neq \mathsf{val}(\varphi_1)$ with $\varphi_2 \in \mathsf{domBy}(\varphi')$ and $\varphi' \in \mathsf{domBy}(\varphi_1)$, we write $\varphi_2 \lhd \varphi_1$ — intuitively, this condition implies that φ_2 is less powerful than φ_1 as we can find a cycle dominating φ_2 that is itself dominated by φ_1. We show that relation $\lhd$ is a strict preorder (which is not total in general).

> **Lemma 5.6.14** *Relation $\lhd$ is a strict preorder.*

Proof. We first prove that $\lhd$ is irreflexive, i.e., that for all $\varphi \in \Phi_{\mathcal{M}}, \varphi \not\lhd \varphi$. If $\varphi \lhd \varphi$, since $\mathsf{val}(\varphi) = \mathsf{val}(\varphi)$, there exists $\varphi' \in \Phi_{\mathcal{M}}$ such that $\mathsf{val}(\varphi') \neq \mathsf{val}(\varphi)$, $\varphi \in \mathsf{domBy}(\varphi')$, and $\varphi' \in \mathsf{domBy}(\varphi)$. But that is not possible since when φ and φ' are competing, they cannot both dominate the other (no matter the choice of witness, as shown in Lemma 5.6.12).

We now prove that $\lhd$ is transitive. We distinguish four cases (we rename cycles in each case to ease the reading by making it so that cycles with a
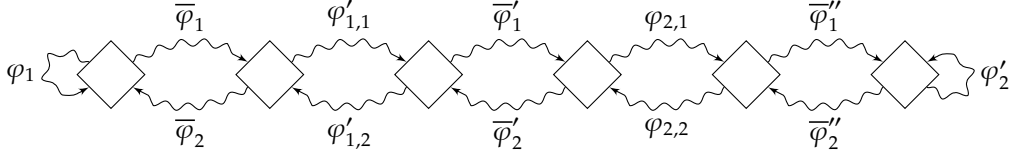
Figure 5.10: Situation to show transitivity of ◁ in Lemma 5.6.14.

prime symbol have a different value from cycles without a prime symbol).

If $\varphi_2 \triangleleft \varphi'$ and $\varphi' \triangleleft \varphi_1$ with $\mathsf{val}(\varphi_2) \neq \mathsf{val}(\varphi')$ and $\mathsf{val}(\varphi') \neq \mathsf{val}(\varphi_1)$, then $\mathsf{val}(\varphi_2) = \mathsf{val}(\varphi_1)$, and $\varphi_2 \triangleleft \varphi_1$ by definition.

Let $\varphi_2' \triangleleft \varphi_2$ and $\varphi_2 \triangleleft \varphi_1$ with $\mathsf{val}(\varphi_2') \neq \mathsf{val}(\varphi_2)$ and $\mathsf{val}(\varphi_2) = \mathsf{val}(\varphi_1)$. We assume w.l.o.g. that $\mathsf{val}(\varphi_2) = \mathsf{val}(\varphi_1) = \mathsf{win}$, so there exists $\varphi_1' \in \Phi_\mathcal{M}$ such that $\mathsf{val}(\varphi_1') = \mathsf{lose}$, $\varphi_2 \triangleleft \varphi_1'$ and $\varphi_1' \triangleleft \varphi_1$. We assume that $\varphi_1' \triangleleft \varphi_1$ is witnessed by $\overline{\varphi}$, that $\varphi_2 \triangleleft \varphi_1'$ is witnessed by $\overline{\varphi}'$ and that $\varphi_2' \triangleleft \varphi_2$ is witnessed by $\overline{\varphi}''$. We assume that $\overline{\varphi} = \overline{\varphi}_1 \overline{\varphi}_2$, $\varphi_1' = \varphi_{1,1}' \varphi_{1,2}'$, $\overline{\varphi}' = \overline{\varphi}_1' \overline{\varphi}_2'$, $\varphi_2 = \varphi_{2,1} \varphi_{2,2}$, and $\overline{\varphi}'' = \overline{\varphi}_1'' \overline{\varphi}_2''$. We refer to Figure 5.10 to illustrate the situation and explain where the common states of these cycles are.

We want to show that $\varphi_2' \triangleleft \varphi_1$. To do so, we show that for $\psi_1 = \overline{\varphi}_1 \varphi_{1,1}' \overline{\varphi}_1' \varphi_{2,1} \overline{\varphi}_1''$ and $\psi_2 = \overline{\varphi}_2'' \varphi_{2,2} \overline{\varphi}_2' \varphi_{1,2}' \overline{\varphi}_2$, $\psi = \psi_1 \psi_2$ is a witness that φ_1 and φ_2' are competing, and that $\varphi_1 \psi_1 \varphi_2' \psi_2$ is winning:

- ▶ Cycle $\varphi_1 \psi$ is winning. We can split this cycle into $\underline{\varphi_{1,2}' \overline{\varphi}_2 \varphi_1 \overline{\varphi}_1 \varphi_{1,1}'}$ and $\overline{\varphi}_1' \varphi_{2,1} \overline{\varphi}_1'' \overline{\varphi}_2'' \varphi_{2,2} \overline{\varphi}_2'$. We shift (Lemma 5.6.3) the former cycle to $\varphi_1 \overline{\varphi}_1 \varphi_{1,1}' \varphi_{1,2}' \overline{\varphi}_2$, which is winning since $\varphi_1' \triangleleft \varphi_1$. The latter cycle has the same value as $\overline{\varphi}_1' (\varphi_{2,2} \varphi_{2,1}) \varphi_{2,1} \overline{\varphi}_1'' \overline{\varphi}_2'' \varphi_{2,2} \overline{\varphi}_2'$ (Corollary 5.6.7), which can be shifted to $(\overline{\varphi}_2' \overline{\varphi}_1' \varphi_{2,2} \varphi_{2,1})(\varphi_{2,1} \overline{\varphi}_1'' \overline{\varphi}_2'' \varphi_{2,2})$. Both these cycles are winning since $\overline{\varphi}'$ and $\overline{\varphi}''$ are witnesses for competitions involving φ_2.
- ▶ Cycle $\varphi_2' \psi$ is losing. We can split this cycle into the two cycles $\overline{\varphi}_1'' \varphi_2' \overline{\varphi}_2''$ and $\varphi_{2,2} \overline{\varphi}_2' \varphi_{1,2}' \overline{\varphi}_2 \overline{\varphi}_1 \varphi_{1,1}' \overline{\varphi}_1' \varphi_{2,1}$. The former is losing because $\overline{\varphi}''$ witnesses a competition involving φ_2'. This latter has the same value as the cycle $\varphi_{2,2} \overline{\varphi}_2' (\varphi_{1,2}' \varphi_{1,1}') \varphi_{1,2}' \overline{\varphi}_2 \overline{\varphi}_1 \varphi_{1,1}' \overline{\varphi}_1' \varphi_{2,1}$ (Corollary 5.6.7), which can itself be split into $\varphi_{1,2}' \overline{\varphi}_2 \overline{\varphi}_1 \varphi_{1,1}'$, which is losing because $\overline{\varphi}$ witnesses a competition involving φ', and $\overline{\varphi}_2' \varphi_{1,2}' \varphi_{1,1}' \overline{\varphi}_1' \varphi_{2,1} \varphi_{2,2}$, which is losing because $\varphi_2 \triangleleft \varphi_1'$.
- ▶ Cycle $\varphi_1 \psi_1 \varphi_2' \psi_2$ is winning. Using Corollary 5.6.7, we can show that $\varphi_1 \psi_1 \varphi_2' \psi_2$ has the same value as $\varphi_1 \psi_1 (\overline{\varphi}_2'' \varphi_{2,2} \varphi_{2,1} \overline{\varphi}_1'') \varphi_2' \psi_2$. We can split this cycle into $\varphi_1 \psi_1 \psi_2 = \varphi_1 \psi$ (which is winning, as shown above) and $\overline{\varphi}_2'' \varphi_{2,2} \varphi_{2,1} \overline{\varphi}_1'' \varphi_2'$ (winning since $\varphi_2' \triangleleft \varphi_2$).

This shows that $\varphi_2' \triangleleft \varphi_1$.

There are still two cases left to consider. The case $\varphi_2' \triangleleft \varphi_1'$ and $\varphi_1' \triangleleft \varphi_1$ with $\mathsf{val}(\varphi_2') = \mathsf{val}(\varphi_1')$ and $\mathsf{val}(\varphi_1') \neq \mathsf{val}(\varphi_1)$ can be dealt with in the same way as the previous case (after noticing that there exists $\varphi_2 \in \Phi_\mathcal{M}$ such that $\varphi_2' \triangleleft \varphi_2$, $\varphi_2 \triangleleft \varphi_1'$ and $\mathsf{val}(\varphi_2) = \mathsf{val}(\varphi_1)$).

If $\varphi_3 \lhd \varphi_2$ and $\varphi_2 \lhd \varphi_1$ with $\mathsf{val}(\varphi_3) = \mathsf{val}(\varphi_2) = \mathsf{val}(\varphi_1)$, then there exists in particular $\varphi' \in \Phi_\mathcal{M}$ such that $\varphi_3 \lhd \varphi'$, $\varphi' \lhd \varphi_2$ and $\mathsf{val}(\varphi_3) \neq \mathsf{val}(\varphi')$. By a previous case, we conclude from $\varphi' \lhd \varphi_2$ and $\varphi_2 \lhd \varphi_1$ that $\varphi' \lhd \varphi_1$. As $\varphi_3 \lhd \varphi'$ and $\varphi' \lhd \varphi_1$, we have $\varphi_3 \lhd \varphi_1$ as desired. □

We define an equivalence relation on the cycles: we write $\varphi_1 \simeq \varphi_2$ if $\mathsf{val}(\varphi_1) = \mathsf{val}(\varphi_2)$, $\mathsf{comp}(\varphi_1) = \mathsf{comp}(\varphi_2)$, and $\mathsf{domBy}(\varphi_1) = \mathsf{domBy}(\varphi_2)$. We show that cycles that are equivalent for $\simeq$ are in relation with the same elements for $\lhd$.

Lemma 5.6.15 *Let $\varphi_1, \varphi_2, \varphi' \in \Phi_\mathcal{M}$.*

- ▶ *If $\varphi_1 \simeq \varphi_2$ and $\varphi' \lhd \varphi_1$, then $\varphi' \lhd \varphi_2$.*
- ▶ *If $\varphi_1 \simeq \varphi_2$ and $\varphi_1 \lhd \varphi'$, then $\varphi_2 \lhd \varphi'$.*

In other words, preorder $\lhd$ is compatible with $\simeq$.

Proof. The first item is straightforward, as the elements smaller than φ_1 for $\lhd$ are determined by $\mathsf{domBy}(\varphi_1)$, and $\mathsf{domBy}(\varphi_1) = \mathsf{domBy}(\varphi_2)$. For the second item, we distinguish two cases:

- ▶ if $\mathsf{val}(\varphi_1) \neq \mathsf{val}(\varphi')$, then $\varphi_1 \lhd \varphi'$ means that $\varphi' \in \mathsf{comp}(\varphi_1)$ and $\varphi' \notin \mathsf{domBy}(\varphi_1)$. If $\varphi_1 \simeq \varphi_2$, the same properties also hold for φ_2, so $\varphi_2 \lhd \varphi'$.
- ▶ if $\mathsf{val}(\varphi_1) = \mathsf{val}(\varphi')$, then $\varphi_1 \lhd \varphi'$ means that there exists φ'' with $\mathsf{val}(\varphi_1) \neq \mathsf{val}(\varphi'')$ such that $\varphi_1 \in \mathsf{domBy}(\varphi'')$ and $\varphi'' \in \mathsf{domBy}(\varphi')$. By the previous case, if $\varphi_1 \simeq \varphi_2$, then $\varphi_2 \lhd \varphi''$, so $\varphi_2 \lhd \varphi'$ as well. □

Partial preorder $\lhd$ therefore also induces a partial preorder on $\Phi_\mathcal{M}/_\simeq$.

Example 5.6.16 We represent the relations between all elements of $\Phi_\mathcal{M}/_\simeq$ for the parity automaton considered in Example 5.6.13 in their *Hasse diagram*, depicted in Figure 5.9 (right). Elements that are linked by a line segment are comparable for $\lhd$, and elements that are above are greater for $\lhd$. There are four equivalence classes of cycles, two of them winning and two of them losing. Notice for instance that any cycle going through transition (m_1, c) is equivalent (for $\simeq$) to cycle (m_1, c): indeed, it is necessarily a losing cycle competing with and dominating all the winning cycles in this memory structure. Other examples are given by $(m_1, a)(m_2, a) \simeq (m_1, a)(m_2, b)(m_2, a)$ and $(m_2, b) \simeq (m_2, c)$.

We now prove finiteness of the index of $\simeq$, by showing that

- ▶ the *height* of partial preorder $\lhd$ is finite, i.e., there is no infinite increasing nor decreasing sequence for $\lhd$ (Lemma 5.6.19);
- ▶ the *width* of partial preorder $\lhd$ on $\Phi_\mathcal{M}/_\simeq$ is finite, i.e., there is no infinite set of elements in $\Phi_\mathcal{M}/_\simeq$ that are all pairwise incomparable for $\lhd$ (Lemma 5.6.20).
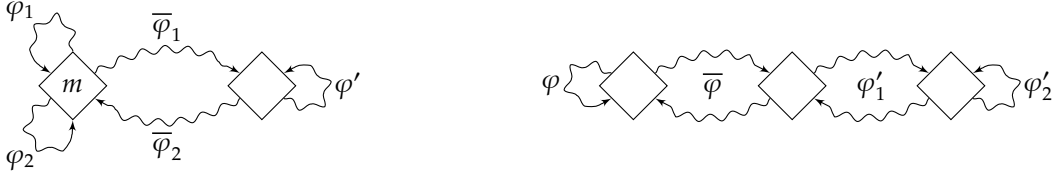
Figure 5.11: Situation in the proof of Lemma 5.6.17 (left) and in the proof of Lemma 5.6.18 (right).

We start with two technical lemmas about competition between cycles.

> **Lemma 5.6.17** *Let $\varphi_1, \varphi_2, \varphi' \in \Phi_{\mathcal{M}}$ be such that $\mathsf{val}(\varphi_1) = \mathsf{val}(\varphi_2) \neq \mathsf{val}(\varphi')$, $\varphi_2 \lhd \varphi'$, and $\varphi' \lhd \varphi_1$. Let $\overline{\varphi}$ be a witness that φ_2 and φ' are competing such that $\mathsf{st}(\overline{\varphi}) \cap \mathsf{st}(\varphi_1) \cap \mathsf{st}(\varphi_2) \neq \emptyset$. Then, $\overline{\varphi}$ also witnesses that φ_1 and φ' are competing.*

Proof. We already know that $\mathsf{val}(\varphi'\overline{\varphi}) = \mathsf{val}(\varphi')$ and that $\mathsf{st}(\varphi') \cap \mathsf{st}(\overline{\varphi}) \neq \emptyset$ (as $\overline{\varphi}$ witnesses a competition involving φ') and that $\mathsf{st}(\varphi_1) \cap \mathsf{st}(\overline{\varphi}) \neq \emptyset$ (by hypothesis). It is left to show that $\mathsf{val}(\varphi_1\overline{\varphi}) = \mathsf{val}(\varphi_1)$. Let $m \in \mathsf{st}(\overline{\varphi}) \cap \mathsf{st}(\varphi_1) \cap \mathsf{st}(\varphi_2)$; we represent the situation in Figure 5.11 (left), with $\overline{\varphi} = \overline{\varphi}_1\overline{\varphi}_2$. Consider first cycle $\overline{\varphi}_1\overline{\varphi}_2\varphi_2$: this cycle is a witness that φ_1 and φ' are competing, since it has common states with those cycles, $\mathsf{val}(\varphi_1(\overline{\varphi}_1\overline{\varphi}_2\varphi_2)) = \mathsf{val}(\varphi_1)$ (both φ_1 and $\overline{\varphi}_1\overline{\varphi}_2\varphi_2$ have the same value), and $\mathsf{val}(\varphi'(\overline{\varphi}_2\varphi_2\overline{\varphi}_1)) = \mathsf{val}(\varphi')$ (since $\varphi_2 \lhd \varphi'$ and $\overline{\varphi}$ is a witness of the competition). Therefore, as $\varphi' \lhd \varphi_1$, the cycle $\psi = \varphi_1\overline{\varphi}_1\varphi'\overline{\varphi}_2\varphi_2$ has the same value as φ_1. By Corollary 5.6.7, cycle ψ has the same value as $\varphi_1(\overline{\varphi}_1\overline{\varphi}_2)\overline{\varphi}_1\varphi'\overline{\varphi}_2\varphi_2$, which can be split into $\overline{\varphi}_1\varphi'\overline{\varphi}_2\varphi_2$ (which has the same value as φ' since $\varphi_2 \lhd \varphi'$ and $\overline{\varphi}$ is a witness of the competition) and $\varphi_1\overline{\varphi}_1\overline{\varphi}_2$. Therefore, $\varphi_1\overline{\varphi}_1\overline{\varphi}_2 = \varphi_1\overline{\varphi}$ cannot have the same value as φ', otherwise ψ would also have the same value as φ' by $\mathcal{M}$-cycle-consistency. □

> **Lemma 5.6.18** *Let $\varphi, \varphi_1' \in \Phi_{\mathcal{M}}$ be such that $\mathsf{val}(\varphi) \neq \mathsf{val}(\varphi_1')$ and $\varphi_1' \lhd \varphi$. Let φ_2' be a cycle such that $\mathsf{val}(\varphi_2') = \mathsf{val}(\varphi_1')$ and $\mathsf{st}(\varphi_2') \cap \mathsf{st}(\varphi_1') \neq \emptyset$. Then, φ and φ_2' are competing.*

Proof. Let $\overline{\varphi}$ be a witness that φ and φ_1' are competing; we represent the situation in Figure 5.11 (right). We show that $\overline{\varphi}\varphi_1'$ is a witness that φ and φ_2' are competing. As $\mathsf{st}(\overline{\varphi}) \cap \mathsf{st}(\varphi) \neq \emptyset$, we have $\mathsf{st}(\overline{\varphi}\varphi_1') \cap \mathsf{st}(\varphi) \neq \emptyset$. Similarly, as $\mathsf{st}(\varphi_1') \cap \mathsf{st}(\varphi_2') \neq \emptyset$, we have $\mathsf{st}(\overline{\varphi}\varphi_1') \cap \mathsf{st}(\varphi_2') \neq \emptyset$. As $\varphi_1' \lhd \varphi$ with witness $\overline{\varphi}$, we have that $\mathsf{val}(\varphi(\overline{\varphi}\varphi_1')) = \mathsf{val}(\varphi)$. Moreover, since $\overline{\varphi}$ is a witness for φ_1' (and φ'), $\mathsf{val}(\overline{\varphi}\varphi_1') = \mathsf{val}(\varphi_1')$. Therefore $\mathsf{val}(\overline{\varphi}\varphi_1') = \mathsf{val}(\varphi_2')$, which implies by $\mathcal{M}$-cycle-consistency that $\mathsf{val}(\varphi_2'(\overline{\varphi}\varphi_1')) = \mathsf{val}(\varphi_2')$. □

> **Lemma 5.6.19** *The height of partial preorder $\lhd$ is finite.*
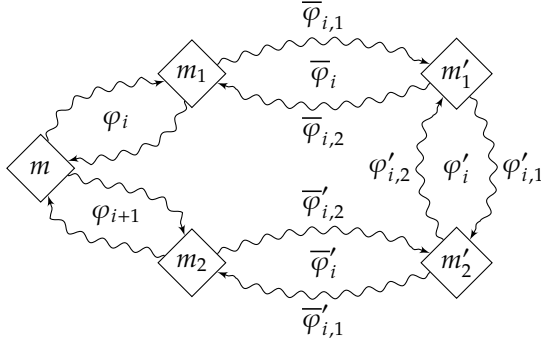
Figure 5.12: Situation in the proof of Lemma 5.6.19. Competition of φ_i and φ'_i is witnessed by $\overline{\varphi}_i$, and competition of φ'_i and φ_{i+1} is witnessed by $\overline{\varphi}'_i$. State m' appears somewhere along φ'_i and is not represented.

Proof. By contradiction, let $\varphi_0 \rhd \varphi'_0 \rhd \varphi_1 \rhd \varphi'_1 \rhd \varphi_2 \rhd \ldots$ be an infinite decreasing sequence for $\lhd$. We assume w.l.o.g. that for all $i \geq 0$, $\mathsf{val}(\varphi_i) = \mathsf{win}$ and $\mathsf{val}(\varphi'_i) = \mathsf{lose}$ (if two consecutive cycles are, for example, both winning, we can always insert an intermediate losing cycle between them, by definition).

For $i \geq 0$, let m (resp. m') be a state of $\mathcal{M}$ that is part of infinitely many sets $\mathsf{st}(\varphi_i)$ (resp. $\mathsf{st}(\varphi'_i)$) — such states necessarily exist as the state space of $\mathcal{M}$ is finite. Thanks to transitivity of $\lhd$ (Lemma 5.6.14), by keeping only winning cycles φ_i such that $m \in \mathsf{st}(\varphi_i)$ alternating with losing cycles φ'_i such that $m' \in \mathsf{st}(\varphi'_i)$, we keep an infinite decreasing sequence for $\lhd$. We can therefore assume w.l.o.g., up to renaming cycles, that for all $i \in \mathbb{N}$, $m \in \mathsf{st}(\varphi_i)$ and $m' \in \mathsf{st}(\varphi'_i)$.

This is the first time in the proofs of this chapter that we use that $\mathcal{M}$ has a finite state space. Therefore, Theorem 5.4.1 does not carry over to memory structures allowed to have infinitely many states. We leave open the existence of a characterization such as ours with infinite memory structures.

We show that the competition of each contiguous pair in sequence $\varphi_0, \varphi'_0, \varphi_1, \varphi'_1, \varphi_2, \ldots$ has a witness that intersects the winning cycle at m, and the losing cycle at m'. For all $i \geq 0$, let $\overline{\varphi}_i = \overline{\varphi}_{i,1}\overline{\varphi}_{i,2}$ be a witness that φ_i and φ'_i are competing, and $\overline{\varphi}'_i = \overline{\varphi}'_{i,1}\overline{\varphi}'_{i,2}$ be a witness that φ'_i and φ_{i+1} are competing. Let $i \geq 0$; we depict part of the situation in Figure 5.12, with $\varphi'_i = \varphi'_{i,1}\varphi'_{i,2}$.

Based on the cycles that we already know, we consider the cycle $\overline{\psi}_i = \overline{\varphi}_{i,1}\varphi'_{i,1}\overline{\varphi}'_{i,1}\varphi_{i+1}\overline{\varphi}'_{i,2}\varphi'_{i,2}\overline{\varphi}_{i,2}$. We have that $m, m' \in \mathsf{st}(\overline{\psi}_i)$ since φ_{i+1} and φ'_i are part of $\overline{\psi}$. We show that $\overline{\psi}_i$ witnesses that φ_i and φ'_i are competing:

- ▶ $\mathsf{val}(\varphi_i\overline{\psi}_i) = \mathsf{win}$ since $\varphi_i\overline{\psi}_i$ can be split into $\varphi'_{i,2}\overline{\varphi}_{i,2}\varphi_i\overline{\varphi}_{i,1}\varphi'_{i,1}$ (winning since $\varphi_i \rhd \varphi'_i$) and $\overline{\varphi}'_{i,1}\varphi_{i+1}\overline{\varphi}'_{i,2}$ (winning since $\overline{\varphi}'_i$ witnesses a competition involving φ_{i+1});
- ▶ $\mathsf{val}(\varphi'_i\overline{\psi}_i) = \mathsf{lose}$ since $\varphi'_i\overline{\psi}_i$ can be split into $\varphi'_i\overline{\varphi}_{i,2}\overline{\varphi}_{i,1}$ (losing since $\overline{\varphi}_i$ witnesses a competition involving φ'_i) and $\varphi'_{i,1}\overline{\varphi}'_{i,1}\varphi_{i+1}\overline{\varphi}'_{i,2}\varphi'_{i,2}$ (losing since $\varphi'_i \rhd \varphi_{i+1}$). We use Remark 5.6.9 in order to write "$\varphi'_i\overline{\psi}_i$".

We can perform a symmetric reasoning to show that the competition of any pair $\varphi'_i, \varphi_{i+1}$, $i \geq 0$, is witnessed by a cycle $\overline{\psi}'_i \in \Phi_{\mathcal{M}}$ such that $m', m \in \mathsf{st}(\overline{\psi}'_i)$.

By Lemma 5.6.17, $\overline{\psi}'_i$ is not only a witness that φ'_i and φ_{i+1} are competing, but also that φ_i and φ'_i are competing (indeed, $\mathsf{val}(\varphi_i) = \mathsf{val}(\varphi_{i+1}) \neq$

$\text{val}(\varphi_i')$, $\varphi_{i+1} \lhd \varphi_i'$, $\varphi_i' \lhd \varphi_i$, $\overline{\psi}_i'$ witnesses that φ_{i+1} and φ_i' are competing, and $m \in \text{st}(\overline{\psi}_i') \cap \text{st}(\varphi_i) \cap \text{st}(\varphi_{i+1}))$.

For $i \geq 0$, we can write $\overline{\psi}_i' = \overline{\psi}_{i,1}' \overline{\psi}_{i,2}'$ with $\overline{\psi}_{i,1}' \in \Pi_{m',m}$ and $\overline{\psi}_{i,2}' \in \Pi_{m,m'}$. We now consider the infinite sequence

$$\xi = \varphi_0 \overline{\psi}_{0,2}' \varphi_0' \overline{\psi}_{0,1}' \varphi_1 \overline{\psi}_{1,2}' \varphi_1' \overline{\psi}_{1,1}' \varphi_2 \ldots$$

Notice that for all $i \geq 0$, $\varphi_i \overline{\psi}_{i,2}' \varphi_i' \overline{\psi}_{i,1}'$ is a winning cycle on m since $\varphi_i \rhd \varphi_i'$; hence $\text{col}^\omega(\xi) \in m^{-1}W$ by $\mathcal{M}$-cycle-consistency. Also, for all $i \geq 0$, $\overline{\psi}_{i,2}' \varphi_i' \overline{\psi}_{i,1}' \varphi_{i+1}$ is a losing cycle on m since $\varphi_i' \rhd \varphi_{i+1}$; hence $\text{col}^\omega(\xi) \in m^{-1}\overline{W}$ by $\mathcal{M}$-prefix-independence and $\mathcal{M}$-cycle-consistency. This is a contradiction.

A proof for infinite increasing sequences can be done in a symmetric way. □

Lemma 5.6.20 *The width of partial preorder $\lhd$ on $\Phi_{\mathcal{M}}/\simeq$ is finite.*

Proof. We recall that $\mathcal{M} = (M, m_{\text{init}}, \alpha_{\text{upd}})$. We will show that any two cycles φ_1 and φ_2 such that $\text{st}(\varphi_1) = \text{st}(\varphi_2)$ are necessarily comparable for $\simeq$ or $\lhd$. This will show that the cardinality of a maximal set of pairwise incomparable (for $\lhd$) elements in $\Phi_{\mathcal{M}}/\simeq$ is necessarily bounded by $2^{|M|}$, which implies that the width of partial preorder $\lhd$ is finite as $|M|$ is finite. Let φ_1 and φ_2 be two cycles such that $\text{st}(\varphi_1) = \text{st}(\varphi_2)$ (we recall that there may be infinitely many transitions in $\mathcal{M}$ if C is infinite, and that two cycles going through the same states may use different transitions and have a different value).

If $\text{val}(\varphi_1) \neq \text{val}(\varphi_2)$, then as φ_1 and φ_2 share a common state, they are competing — we have either $\varphi_1 \lhd \varphi_2$ or $\varphi_2 \lhd \varphi_1$ (depending on the value of $\varphi_1\varphi_2$).

We now assume that $\text{val}(\varphi_1) = \text{val}(\varphi_2)$; we assume w.l.o.g. that φ_1 and φ_2 are winning. If φ_1 and φ_2 are such that $\text{comp}(\varphi_1) = \text{comp}(\varphi_2)$, then they can necessarily be compared with $\simeq$ or $\lhd$; indeed,

- ▶ if $\text{domBy}(\varphi_1) = \text{domBy}(\varphi_2)$, then $\varphi_1 \simeq \varphi_2$;
- ▶ if $\text{domBy}(\varphi_1) \neq \text{domBy}(\varphi_2)$, then there is $i \in \{1, 2\}$ and a losing cycle φ' in $\text{domBy}(\varphi_i)$ that is competing with φ_{3-i} but that is not an element of $\text{domBy}(\varphi_{3-i})$. Therefore, $\varphi_{3-i} \lhd \varphi' \lhd \varphi_i$, which means that $\varphi_{3-i} \lhd \varphi_i$.

It is left to deal with the case $\text{comp}(\varphi_1) \neq \text{comp}(\varphi_2)$. W.l.o.g., let φ' be in $\text{comp}(\varphi_1) \setminus \text{comp}(\varphi_2)$. There are two cases to discuss: whether $\varphi_1 \lhd \varphi'$ or $\varphi' \lhd \varphi_1$.

- ▶ Assume $\varphi_1 \lhd \varphi'$. By Lemma 5.6.18, as $\text{val}(\varphi_1) = \text{val}(\varphi_2)$ and $\text{st}(\varphi_1) \cap \text{st}(\varphi_2) \neq \emptyset$, φ' is also competing with φ_2, which is a contradiction.

▶ Assume $\varphi' \lhd \varphi_1$. Let $\overline{\varphi}$ be a witness that φ_1 and φ' are competing. We therefore have that $\varphi_1\overline{\varphi}$ is winning and $\varphi'\overline{\varphi}$ is losing. As φ_2 is not competing with φ', $\overline{\varphi}$ cannot be a witness that φ_2 and φ' are competing. Since $\mathsf{st}(\varphi_1) = \mathsf{st}(\varphi_2)$ has a non-empty intersection with $\mathsf{st}(\overline{\varphi})$, the only possibility for that to happen is that $\varphi_2\overline{\varphi}$ is losing (all other conditions are satisfied). This means that $\overline{\varphi}$ must itself be a losing cycle. But then, observe that $\overline{\varphi}$ is competing both with φ_1 and φ_2 (as $\overline{\varphi}$ has a common state with and a different value than φ_1 and φ_2) and $\varphi_2 \lhd \overline{\varphi} \lhd \varphi_1$ (as $\varphi_2\overline{\varphi}$ is losing and $\varphi_1\overline{\varphi}$ is winning). This implies that $\varphi_2 \lhd \varphi_1$. □

Lemmas 5.6.19 and 5.6.20 imply together that $\simeq$ has a finite index, and thus that $\lhd$ (partially) orders only finitely many classes of cycles in $\Phi_{\mathcal{M}}/{\simeq}$. Therefore, for some $n \in \mathbb{N}$, there exists a function $p_\Phi \colon \Phi_{\mathcal{M}}/{\simeq} \to \{0, \ldots, n\}$ that is a monotonic function (assuming $\Phi_{\mathcal{M}}/{\simeq}$ is preordered with $\lhd$ and $\{0, \ldots, n\}$ is ordered with the usual order on $\mathbb{N}$); such a function is sometimes called a *linear extension of the partial order*. We extend it to a function $p_\Phi \colon \Phi_{\mathcal{M}} \to \{0, \ldots, n\}$ such that $p_\Phi(\varphi) = p_\Phi([\varphi]_{\simeq})$. Moreover, we assume w.l.o.g. that $\mathsf{val}(\varphi) = \mathsf{win}$ if and only if $p_\Phi(\varphi)$ is even (this might require increasing n).

We fix n and any such function p_Φ for the rest of the proof.

5.6.7 Parity automaton on top of $\mathcal{M}$

At this point, it would already be possible to describe words of W in terms of the cycles of $\mathcal{M}$ that they visit (there may be multiple such decompositions) and their values by p_Φ, but that does not directly correspond to a classical acceptance condition for automata on infinite words. We can actually obtain something more satisfying: we show that we can assign priorities to *transitions* of $\mathcal{M}$ to recognize W, in a way that corresponds to a parity acceptance condition on transitions. We transfer function p_Φ to transitions of $\mathcal{M}$: for $(m, c) \in M \times C$, we define

$$p(m, c) = \min\{p_\Phi(\varphi) \mid \varphi \in \Phi_{\mathcal{M}}, (m, c) \in \varphi\}. \tag{5.1}$$

We now have a well-defined function assigning priorities to every transition of $\mathcal{M}$.

Example 5.6.21 We illustrate our definitions for p_Φ and p. We again consider the example from Figure 5.9 (for which, unlike W, we already know that it describes an ω-regular objective). For the sake of the example, let us ignore the already-defined priority function p of this parity automaton. We show that we can recover priorities defining the same language starting from our preorder $\lhd$ and our definitions for p_Φ and p. There were four equivalence classes for $\simeq$, ordered as follows: $[(m_1, b)]_{\simeq} \lhd [(m_1, a)(m_2, a)]_{\simeq} \lhd [(m_1, c)]_{\simeq}$ and $[(m_2, b)]_{\simeq} \lhd [(m_1, c)]_{\simeq}$.

Function p_Φ must be any function that respects the order given by the diagram and that assigns even priorities to winning classes of cycles, and odd priorities to losing classes. One such possible choice is $p_\Phi([(m_1, c)]_\simeq) = 5$, $p_\Phi([(m_1, a)(m_2, a)]_\simeq) = 2$, $p_\Phi([(m_2, b)]_\simeq) = 4$, and $p_\Phi([(m_1, b)]_\simeq) = 1$. From this choice of function p_Φ, our definition of function p (Equation (5.1)) entails $p(m_1, c) = 5$, $p(m_1, a) = p(m_2, a) = 2$, $p(m_2, b) = p(m_2, c) = 4$, and $p(m_1, b) = 1$. This choice of priorities recognizes the same objective as the original parity automaton.

We prove that DPA $(\mathcal{M}, p)$ recognizes the objective W. In our proof, we will need to relate the cycles dominated by a cycle φ and the ones dominated by cycles in a "decomposition" of φ, i.e., cycles that can be obtained from iteratively removing cycles from φ. We formally define this notion and prove two related results.

Definition 5.6.22 *Let $\varphi = (m_0, c_1) \ldots (m_{k-1}, c_k) \in \Phi_\mathcal{M}$, and $\varphi_1, \ldots, \varphi_l \in \Phi_\mathcal{M}$. We say that $(\varphi_1, \ldots, \varphi_l)$ is a* cycle decomposition *of φ if*

- ▶ *either $l = 1$ and $\varphi = \varphi_1$,*
- ▶ *or $l > 1$ and there exist $i, i' \in \{0, \ldots, k-1\}, i \leq i'$, such that cycle $\varphi_1 = (m_i, c_{i+1}) \ldots (m_{i'}, c_{i'+1})$, and $(\varphi_2, \ldots, \varphi_l)$ is a cycle decomposition of the smaller cycle $(m_0, c_1) \ldots (m_{i-1}, c_i)(m_{i'+1}, c_{i'+2}) \ldots (m_{k-1}, c_k)$.*

Lemma 5.6.23 *Let $\varphi, \varphi_1, \varphi_2, \varphi' \in \Phi_\mathcal{M}$ be cycles such that $\varphi = \varphi_1\varphi_2$. If $\varphi' \lhd \varphi_1$, then $\varphi' \lhd \varphi$.*

Proof. We assume $\varphi' \lhd \varphi_1$.

If $\mathsf{val}(\varphi_1) \neq \mathsf{val}(\varphi)$, then $\varphi_1 \lhd \varphi$ — indeed, they share at least one state and $\varphi_1\varphi = (\varphi_1)^2\varphi_2$ has the same value as $\varphi_1\varphi_2 = \varphi$ by Lemma 5.6.5. Therefore, by transitivity of $\lhd$ (Lemma 5.6.14), $\varphi' \lhd \varphi$.

We recall that we use Remark 5.6.9 to write $\varphi_1\varphi$ and $(\varphi_1)^2\varphi_2$ without worrying about the memory state used to go from one to the other.

We now assume $\mathsf{val}(\varphi_1) = \mathsf{val}(\varphi)$ and $\mathsf{val}(\varphi') \neq \mathsf{val}(\varphi_1)$. Let $\overline{\varphi}$ be a witness that φ' and φ_1 are competing. We prove that $\overline{\varphi}$ also witnesses that φ' and φ are competing: to do so, it is left to show that $\psi = \overline{\varphi}\varphi$ has the same value as φ. We have that ψ can be written as $\overline{\varphi}\varphi_{1,1}\varphi_2\varphi_{1,2}$ for some paths $\varphi_{1,1}$ and $\varphi_{1,2}$ such that $\varphi_1 = \varphi_{1,1}\varphi_{1,2}$. Cycle ψ has the same value as $\overline{\varphi}(\varphi_{1,1}\varphi_{1,2})\varphi_{1,1}\varphi_2\varphi_{1,2}$ by Corollary 5.6.7. This last cycle can be split into $\overline{\varphi}\varphi_1$ and φ, which both have the same value as φ. Therefore $\overline{\varphi}$ is also a witness that φ' and φ are competing. We can show with a very similar argument that $\varphi'\overline{\varphi}_1\varphi\overline{\varphi}_2$ also has the same value as φ, hence $\varphi' \lhd \varphi$.

If $\mathsf{val}(\varphi_1) = \mathsf{val}(\varphi)$ and $\mathsf{val}(\varphi') = \mathsf{val}(\varphi_1)$, then there exists φ'' with $\mathsf{val}(\varphi'') \neq \mathsf{val}(\varphi_1)$ such that $\varphi' \in \mathsf{domBy}(\varphi'')$ and $\varphi'' \in \mathsf{domBy}(\varphi_1)$, so $\varphi' \lhd \varphi'' \lhd \varphi_1$. By the previous case, $\varphi'' \lhd \varphi$; by transitivity, $\varphi' \lhd \varphi$. □

Lemma 5.6.24 *Let φ be a cycle of $\mathcal{M}$ and $(\varphi_1, \ldots, \varphi_l)$ be a cycle decomposition of φ. For all $i \in \{1, \ldots, l\}$, for all $\varphi' \in \Phi_{\mathcal{M}}$, if $\varphi' \lhd \varphi_i$, then $\varphi' \lhd \varphi$.*

Proof. We proceed by induction on l. If $l = 1$, then the statement is trivial as $\varphi = \varphi_1$. For $l > 1$, we now assume that the property holds for $l - 1$, and we show that it also holds for l. Up to a shift of φ and of the cycle decomposition, we assume that φ is equal to $\varphi_1 \psi$, where $(\varphi_2, \ldots, \varphi_l)$ is a cycle decomposition of ψ.

Let $\varphi' \in \Phi_{\mathcal{M}}$ be such that $\varphi' \lhd \varphi_i$ for some $i \in \{1, \ldots, l\}$. This implies that $\varphi' \lhd \varphi_1$ if $i = 1$ or, using the induction hypothesis, that $\varphi' \lhd \psi$. In any case, by Lemma 5.6.23, we immediately have that $\varphi' \lhd \varphi$. $\qquad\square$

We can now prove that W is recognized by the parity automaton $(\mathcal{M}, p)$. We do this in the next two results. First, we show that winning cycles of $\mathcal{M}$ are exactly the ones that have an even maximal priority given by p. It is then straightforward to conclude that infinite words in W are exactly the ones whose maximal infinitely visited priority is even.

Lemma 5.6.25 *Let $\varphi = (m_0, c_1) \ldots (m_{k-1}, c_k) \in \Phi_{\mathcal{M}}$. Then, φ is winning if and only if $\max_{0 \le i < k} p(m_i, c_{i+1})$ is even.*

Proof. For conciseness, let $p^* = \max_{0 \le i < k} p(m_i, c_{i+1})$ and $t_i = (m_i, c_{i+1})$. By definition of function p, for all $i \in \{0, \ldots, k-1\}$, $p(t_i) \le p_{\Phi}(\varphi)$. Hence, $p^* \le p_{\Phi}(\varphi)$. We want to show that φ is winning if and only if p^* is even. By contradiction, we assume that we do not have this equivalence. We assume w.l.o.g. that φ is losing and that p^* is even; we could obtain in a symmetric way a contradiction for φ winning and p^* odd.

As φ is losing, we have that $p_{\Phi}(\varphi)$ is odd — as p^* is even, $p^* < p_{\Phi}(\varphi)$. We assume (up to a shift of the transitions) that $p^* = p(t_0)$. Since $p^* < p_{\Phi}(\varphi)$, there exists, for all $i \in \{0, \ldots, k-1\}$, a cycle $\varphi_i \ne \varphi$ such that $t_i \in \varphi_i$ and $p(t_i) = p_{\Phi}(\varphi_i)$. We assume $\varphi_i = t_i \pi_i$ for a suitable path π_i. The situation is represented in Figure 5.13.

The rest of the proof consists in exhibiting two cycles, building on the ones we know, showing that one of them is winning and one of them is losing, and finally showing that they must have the same value, which provides a contradiction.

We will first consider cycle $t_0 \ldots t_{k-1} \pi_{k-1} \ldots \pi_0$ on m_0. We prove by induction that it is winning. First, $t_0 \pi_0$ is winning since $p_{\Phi}(t_0 \pi_0) = p^*$ is even. Assume now that for $0 < l < k$, $t_0 \ldots t_{l-1} \pi_{l-1} \ldots \pi_0$ is winning. We show that $t_0 \ldots t_{l-1}(t_l \pi_l) \pi_{l-1} \ldots \pi_0$ is winning.

- ▶ If $t_l \pi_l$ is a winning cycle, it follows from $\mathcal{M}$-cycle-consistency.
- ▶ If $t_l \pi_l$ is a losing cycle, we distinguish two cases.
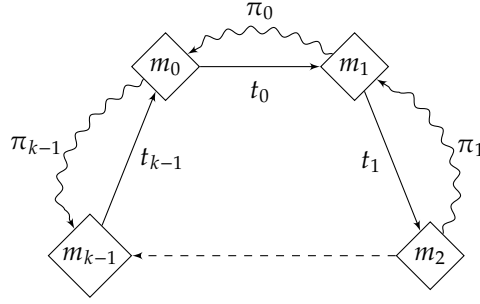
Figure 5.13: Situation in the proof of Lemma 5.6.25, with $\varphi = t_1 \ldots t_k$.

- If $t_1 \ldots t_{l-1}(t_l\pi_l)\pi_{l-1} \ldots \pi_1$ is winning, then so is the cycle $t_0 \ldots t_{l-1}(t_l\pi_l)\pi_{l-1} \ldots \pi_0$ because we just concatenate the winning cycle $\pi_0 t_0$ to a winning cycle ($\mathcal{M}$-cycle-consistency).

- If $t_1 \ldots t_{l-1}(t_l\pi_l)\pi_{l-1} \ldots \pi_1$ is losing, then $t_1 \ldots t_{l-1}\pi_{l-1} \ldots \pi_1$ witnesses that $t_0\pi_0$ and $t_l\pi_l$ are competing. Since $p_\Phi(t_l\pi_l)$ is odd, and $p_\Phi(t_0\pi_0)$ is even and is equal to the maximum of $i \mapsto p_\Phi(t_i\pi_i)$, we have that $p_\Phi(t_l\pi_l) < p_\Phi(t_0\pi_0)$. Since p_Φ is monotonic and $t_0\pi_0$ and $t_l\pi_l$ are competing, this implies $t_l\pi_l \lhd t_0\pi_0$. Thus $t_0 \ldots t_{l-1}(t_l\pi_l)\pi_{l-1} \ldots \pi_0$ is winning.

We now consider the cycle $t_0(\pi_0t_0) \ldots t_{k-1}(\pi_{k-1}t_{k-1})$ on m_0. We show by induction that it is losing. We start from φ, which is losing by hypothesis, and we add cycles $\pi_i t_i$ one by one. We denote $\varphi^{(l)} = t_0(\pi_0t_0) \ldots t_l(\pi_lt_l)t_{l+1} \ldots t_{k-1}$. Assume that $\varphi^{(l-1)}$ is losing for $0 \le l < k$ (we assume $\varphi^{(-1)} = \varphi$). We want to show that $\varphi^{(l)}$ is also losing.

▶ If $\pi_l t_l$ is a losing cycle, it follows from $\mathcal{M}$-cycle-consistency.

▶ If $\pi_l t_l$ is a winning cycle, then as $p_\Phi(\pi_lt_l) \le p^* < p_\Phi(\varphi)$ and $\pi_l t_l$ is competing with φ (they share common states), we have $\pi_l t_l \lhd \varphi$. Notice that $(\pi_0t_0, \ldots, \pi_{l-1}t_{l-1}, \varphi)$ is a cycle decomposition of $\varphi^{(l-1)}$ as in Definition 5.6.22. Thus by Lemma 5.6.24, as $\pi_l t_l \lhd \varphi$, we also have $\pi_l t_l \lhd \varphi^{(l-1)}$. We conclude that $\varphi^{(l)}$ is also losing.

We have considered two cycles on m_0: the winning $t_0 \ldots t_{k-1}\pi_{k-1} \ldots \pi_0$ and the losing $t_0(\pi_0t_0) \ldots t_{k-1}(\pi_{k-1}t_{k-1})$. We show that it is possible to transform the latter into the former using only value-preserving transformations (given by Lemma 5.6.5 and Corollary 5.6.6), which provides the desired contradiction.

We show inductively that for all $l \in \{0, \ldots, k-1\}$, cycle

$$t_0(\pi_0t_0) \ldots t_{k-1}(\pi_{k-1}t_{k-1})$$

can be transformed into

$$\psi^{(l)} = (t_0 \ldots t_l\pi_l \ldots \pi_0t_0 \ldots t_l)t_{l+1}(\pi_{l+1}t_{l+1}) \ldots t_{k-1}(\pi_{k-1}t_{k-1})$$

using value-preserving transformations. Notice that $\psi^{(0)}$ is equal to $t_0(\pi_0t_0) \ldots t_{k-1}(\pi_{k-1}t_{k-1})$, which deals with the base case of the induction. Now assume that $t_0(\pi_0t_0) \ldots t_{k-1}(\pi_{k-1}t_{k-1})$ has the same value as $\psi^{(l-1)}$

for $1 \leq l < k$. In the expression of $\psi^{(l-1)}$, notice that $\pi_{l-1} \ldots \pi_0 t_0 \ldots t_{l-1}$ and $t_l \pi_l$ are two consecutive cycles on m_l. By Lemma 5.6.6, they can thus be swapped while keeping a cycle with the same value. Notice that this gives exactly the cycle $\psi^{(l)}$.

We obtain that $t_0(\pi_0 t_0) \ldots t_{k-1}(\pi_{k-1} t_{k-1})$ has the same value as

$$\psi^{(k-1)} = t_0 \ldots t_{k-1}\pi_{k-1} \ldots \pi_0 t_0 \ldots t_{k-1},$$

which has the same value as $t_0 \ldots t_{k-1}\pi_{k-1} \ldots \pi_0$ by Lemma 5.6.5. This is the desired contradiction. □

Proposition 5.6.26 *Let* $w = c_1 c_2 \ldots \in C^\omega$ *with* $\varrho = (m_0, c_1)(m_1, c_2) \ldots \in (M \times C)^\omega$ *being the run of* $\mathcal{M}$ *on* w. *Then,*

$$w \in W \text{ if and only if } \limsup_{i \to \infty} p(m_i, c_{i+1}) \text{ is even.}$$

Proof. Let $p^* = \limsup_{i \to \infty} p(m_i, c_{i+1})$. Let $j \geq 0$ be an index such that for all $i \geq j$, $p(m_i, c_{i+1}) \leq p^*$. Let $I^* = \{i \geq j \mid p(m_i, c_{i+1}) = p^*\}$ be the infinite set of indices of transitions with priority p^* occurring after index j. We write $i_1, i_2, \ldots$ for the elements of I^* in order. Let m^* be a state appearing infinitely often in $\{m_i \mid i \in I^*\}$ (such a state exists necessarily as the state space of $\mathcal{M}$ is finite). This implies that ϱ can be written as the concatenation of a finite prefix $(m_0, c_1) \ldots (m_{i_1-1}, c_{i_1})$ and infinitely many cycles $\varphi_k = (m_{i_k}, c_{i_k+1}) \ldots (m_{i_{k+1}-1}, c_{i_{k+1}})$ with $m_{i_k} = m^*$ and $p(m_{i_k}, c_{i_k+1}) = p^*$, for $k \geq 1$.

For all $k \geq 1$, we have that $\max_{i_k \leq i < i_{k+1}} p(m_i, c_{i+1}) = p^*$ (it is $\leq p^*$ as $i_k \geq j$, and it is $\geq p^*$ as $p(m_{i_k}, c_{i_k+1}) = p^*$). By Lemma 5.6.25, we conclude that cycles φ_k are all cycles on m^* that have the same value: they are winning if p^* is even, and losing if p^* is odd. By $\mathcal{M}$-prefix-independence and $\mathcal{M}$-cycle-consistency, w is in W if p^* is even, and w is in $\overline{W}$ if p^* is odd. □

We have therefore reached our goal for this section.

Corollary 5.6.27 (Second item of Theorem 5.4.1) *If* W *is* $\mathcal{M}$-*prefix-independent and* $\mathcal{M}$-*cycle-consistent, then* W *is* ω-*regular and can be recognized by a deterministic parity automaton built on top of* $\mathcal{M}$.

Remark 5.6.28 As discussed in Remark 2.7.7, our proof shows as a by-product that even if C is infinite, many colors can be assumed to be equal (w.r.t. W) — there are only finitely many classes of truly different colors.

5.7 Applications

We provide a thorough application of our concepts to a discounted-sum objective. We then discuss more briefly mean-payoff and total-payoff objectives.

5.7.1 Discounted sum

We apply our results to a *discounted-sum* objective in order to illustrate our notions. A specificity of this example is that its ω-regularity depends on some chosen parameters — we use our results to characterize the parameters for which it is ω-regular or, equivalently (Theorem 5.4.3), chromatic-finite-memory-determined. The ω-regularity of discounted-sum objectives has also been studied in [CDH09; BCV18] with different techniques and goals.

In this first part, we strive to illustrate as well as possible the notions and theorems from this chapter. We therefore defer more numerical and elementary proofs to Subsection 5.7.2.

Let $C \subseteq \mathbb{Q}$ be non-empty and bounded. For $\lambda \in (0,1) \cap \mathbb{Q}$, we define the *discounted-sum function* $\mathsf{DS}_\lambda \colon C^\omega \to \mathbb{R}$ such that for $w = c_1 c_2 \ldots \in C^\omega$,

$$\mathsf{DS}_\lambda(w) = \sum_{i=1}^{\infty} \lambda^{i-1} \cdot c_i.$$

This function is always well-defined for a bounded C, and takes values in $\left[\frac{\inf C}{1-\lambda}, \frac{\sup C}{1-\lambda}\right]$.

We define the objective $\mathsf{DS}_\lambda^{\geq 0} = \{w \in C^\omega \mid \mathsf{DS}_\lambda(w) \geq 0\}$ as the set of infinite words having a non-negative discounted sum, and let $\sim$ be its right congruence. We will analyze cycle-consistency and prefix-independence of $\mathsf{DS}_\lambda^{\geq 0}$ to conclude under which conditions (on C and λ) it is chromatic-finite-memory-determined. First, we discuss a few properties of the discounted-sum function.

Basic properties. We extend function DS_λ to finite words in a natural way: for $w \in C^*$, we define $\mathsf{DS}_\lambda(w) = \mathsf{DS}_\lambda(w0^\omega)$. For $w \in C^*$, we define $|w|$ as the length of w (so $w \in C^{|w|}$). First, we notice that for $w \in C^*$ and $w' \in C^\omega$, we have

$$\mathsf{DS}_\lambda(ww') = \mathsf{DS}_\lambda(w) + \lambda^{|w|}\mathsf{DS}_\lambda(w').$$

Therefore,

$$ww' \in \mathsf{DS}_\lambda^{\geq 0} \iff \frac{\mathsf{DS}_\lambda(w)}{\lambda^{|w|}} \geq -\mathsf{DS}_\lambda(w').$$

[CDH09]: Chatterjee et al. (2009), *Expressiveness and Closure Properties for Quantitative Languages*
[BCV18]: Bansal et al. (2018), *Comparator automata in quantitative verification*

We remind the reader that for $\lambda \in (0,1)$,

$$\sum_{i=1}^{\infty} \lambda^{i-1} = \frac{1}{1-\lambda}.$$

This provides a characterization of the winning continuations of a finite word $w \in C^*$ by comparing the discounted sum of the continuations to the value $\frac{\mathsf{DS}_\lambda(w)}{\lambda^{|w|}}$.

This leads us to define the *gap* of a finite word $w \in C^*$, following ideas in [BHO15], as

$$\mathsf{gap}(w) = \begin{cases} \top & \text{if } \frac{\mathsf{DS}_\lambda(w)}{\lambda^{|w|}} \geq -\frac{\inf C}{1-\lambda}, \\ \bot & \text{if } \frac{\mathsf{DS}_\lambda(w)}{\lambda^{|w|}} < -\frac{\sup C}{1-\lambda}, \\ \frac{\mathsf{DS}_\lambda(w)}{\lambda^{|w|}} & \text{otherwise.} \end{cases}$$

Intuitively, the gap of a finite word $w \in C^*$ represents how far its discounted sum is from going back to 0: if $w' \in C^\omega$ is such that $\mathsf{DS}_\lambda(w') = -\mathsf{gap}(w)$, then $\mathsf{DS}_\lambda(ww') = 0$. We can see that for all words $w \in C^*$, if $\mathsf{gap}(w) = \top$, then all continuations are winning (i.e., $w^{-1}W = C^\omega$) as it is not possible to find an infinite word with a discounted sum less than $\frac{\inf C}{1-\lambda}$. Similarly, if $\mathsf{gap}(w) = \bot$, then all continuations are losing (i.e., $w^{-1}W = \emptyset$).

Cycle-consistency. We can show that objective $\mathsf{DS}_\lambda^{\geq 0}$ is always $\mathcal{M}_{\mathsf{triv}}$-cycle-consistent.

> **Proposition 5.7.1** *For all bounded $C \subseteq \mathbb{Q}$, $\lambda \in (0,1) \cap \mathbb{Q}$, objective $\mathsf{DS}_\lambda^{\geq 0}$ is $\mathcal{M}_{\mathsf{triv}}$-cycle-consistent.*

Proof. Let $w \in C^*$. We show that $(\mathsf{col}^*(\Phi_{\mathcal{M}_{\mathsf{triv}}}^{\mathsf{lose},w}))^\omega \subseteq w^{-1}\overline{\mathsf{DS}_\lambda^{\geq 0}}$ — we discuss how to adapt the proof to show that $(\mathsf{col}^*(\Phi_{\mathcal{M}_{\mathsf{triv}}}^{\mathsf{win},w}))^\omega \subseteq w^{-1}\mathsf{DS}_\lambda^{\geq 0}$ at the end. Let $w_1, w_2, \ldots \in \mathsf{col}^*(\Phi_{\mathcal{M}_{\mathsf{triv}}}^{\mathsf{lose},w})$. We want to show that $ww_1w_2\ldots \in \overline{\mathsf{DS}_\lambda^{\geq 0}}$, i.e., that $\mathsf{DS}_\lambda(ww_1w_2\ldots) < 0$.

For $k \geq 1$, as $w_k \in \Phi_{\mathcal{M}_{\mathsf{triv}}}^{\mathsf{lose},w}$, we have $\mathsf{DS}_\lambda(ww_k^\omega) < 0$. Since, moreover,

$$\mathsf{DS}_\lambda(ww_k^\omega) = \mathsf{DS}_\lambda(w) + \lambda^{|w|}\mathsf{DS}_\lambda(w_k^\omega)$$

$$= \mathsf{DS}_\lambda(w) + \lambda^{|w|}\sum_{i=0}^\infty \lambda^{i|w_k|}\mathsf{DS}_\lambda(w_k)$$

$$= \mathsf{DS}_\lambda(w) + \lambda^{|w|}\mathsf{DS}_\lambda(w_k)\frac{1}{1-\lambda^{|w_k|}},$$

we obtain

$$\mathsf{DS}_\lambda(w_k) < -\mathsf{DS}_\lambda(w)\frac{1-\lambda^{|w_k|}}{\lambda^{|w|}}. \tag{5.2}$$

In particular, for $k = 1$, there exists $\epsilon > 0$ such that

$$\mathsf{DS}_\lambda(w_1) = -\epsilon - \mathsf{DS}_\lambda(w)\frac{1-\lambda^{|w_1|}}{\lambda^{|w|}}. \tag{5.3}$$
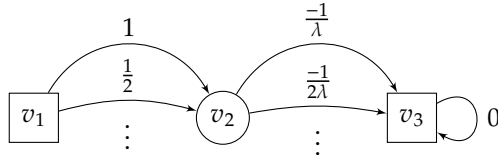
We have that

$$\mathsf{DS}_\lambda(ww_1w_2\ldots) = \mathsf{DS}_\lambda(w) + \lambda^{|w|} \sum_{k=1}^{\infty} \lambda^{\sum_{i=1}^{k-1}|w_i|} \mathsf{DS}_\lambda(w_k)$$

$$\leq \mathsf{DS}_\lambda(w) - \lambda^{|w|}\epsilon - \lambda^{|w|} \sum_{k=1}^{\infty} \lambda^{\sum_{i=1}^{k-1}|w_i|} \mathsf{DS}_\lambda(w)\frac{1 - \lambda^{|w_k|}}{\lambda^{|w|}}$$

$$\leq \mathsf{DS}_\lambda(w) - \lambda^{|w|}\epsilon - \mathsf{DS}_\lambda(w) \sum_{k=1}^{\infty} \lambda^{\sum_{i=1}^{k-1}|w_i|}(1 - \lambda^{|w_k|}),$$

where the second line uses Equation (5.3) for $k = 1$, and Equation (5.2) for $k \geq 2$. The series

$$\sum_{k=1}^{\infty} \lambda^{\sum_{i=1}^{k-1}|w_i|}(1 - \lambda^{|w_k|}) = \sum_{k=1}^{\infty} \lambda^{\sum_{i=1}^{k-1}|w_i|} - \lambda^{\sum_{i=1}^{k}|w_i|}$$

is telescoping (we can expand it as $1 - \lambda^{|w_1|} + \lambda^{|w_1|} - \lambda^{|w_1|+|w_2|} + \lambda^{|w_1|+|w_2|} - \ldots$). As $\lim_{k \to \infty} \lambda^{\sum_{i=1}^{k}|w_i|} = 0$, this series converges to 1. We conclude that

$$\mathsf{DS}_\lambda(ww_1w_2\ldots) \leq \mathsf{DS}_\lambda(w) - \lambda^{|w|}\epsilon - \mathsf{DS}_\lambda(w) = \lambda^{|w|}\epsilon < 0,$$

as required.

A proof that $(\mathsf{col}^*(\Phi_{\mathcal{M}_{\mathsf{triv}}}^{\mathsf{win},\overline{w}}))^\omega \subseteq \overline{w}^{-1}\mathsf{DS}_\lambda^{\geq 0}$ can be done in a similar way, with no need to extract an ϵ as we are then only looking for a non-strict inequality. $\qquad\square$

As $\mathsf{DS}_\lambda^{\geq 0}$ is $\mathcal{M}_{\mathsf{triv}}$-cycle-consistent, it is ω-regular if and only if it is additionally $\mathcal{M}$-prefix-independent for some memory structure $\mathcal{M}$.

Prefix-independence. If $C = [-k, k] \cap \mathbb{Q}$ for some $k \in \mathbb{N} \setminus \{0\}$, objective $\mathsf{DS}_\lambda^{\geq 0}$ is not $\mathcal{M}$-prefix-independent for any $\mathcal{M}$, as $\sim$ has infinite index. Indeed, for $i \geq 1$ and $w_i = \frac{1}{i} \in C^*$, we have $w_1 > w_2 > \ldots$ — we can see how to use this to exhibit an arena in which $\mathcal{P}_1$ can win but needs infinite memory to do so in Figure 5.14.

For finite $C \subseteq \mathbb{Z}$, the picture is more complicated; for $C = [-k, k] \cap \mathbb{Z}$ for some $k \in \mathbb{N}$, we characterize when $\mathsf{DS}_\lambda^{\geq 0}$ is $\mathcal{M}$-prefix-independent for some finite structure $\mathcal{M}$. We give an intuition of the two situations in which that happens: (*i*) if C is too small, then the first non-zero color seen determines the outcome of the game, as it is not possible to compensate this color to change the sign of the discounted sum; (*ii*) if $\lambda = \frac{1}{n}$ for some

integer $n \geq 1$, then the gap function actually takes only finitely many values, which is not the case for other values of λ.

In the proof, we respectively use symbols $\lceil \cdot \rceil$ and $\lfloor \cdot \rfloor$ for the ceiling and floor functions.

> **Proposition 5.7.2** *Let $\lambda \in (0, 1) \cap \mathbb{Q}$, $k \in \mathbb{N}$, and $C = [-k, k] \cap \mathbb{Z}$. Then, the right congruence $\sim$ of $\mathsf{DS}_\lambda^{\geq 0}$ has a finite index if and only if $k < \frac{1}{\lambda} - 1$ or λ is equal to $\frac{1}{n}$ for some integer $n \geq 1$.*

Proof. We define $\mathsf{maxDS} = \frac{\sup C}{1-\lambda} = \frac{k}{1-\lambda}$ and $\mathsf{minDS} = -\frac{k}{1-\lambda}$, as respectively the maximal and minimal discounted-sum value achievable with colors in C.

The key property that we will show is that gaps characterize equivalence classes of prefixes: for $w_1, w_2 \in C^*$,

$$w_1 \sim w_2 \iff \mathsf{gap}(w_1) = \mathsf{gap}(w_2). \tag{5.4}$$

Once this is proven, it is left to determine the number of different gap values, which will correspond to the index of $\sim$. The right-to-left implication of (5.4) is clear from the definition of gap and the related discussion: if the gaps are $\top$, all the continuations are winning; if the gaps are $\bot$, all the continuations are losing; else, for any continuation, the final discounted-sum values will have the same sign. We prove the left-to-right implication for each case of the disjunction from the statement and discuss the number of gap values.

We first assume $k < \frac{1}{\lambda} - 1$. The case $k = 0$ is trivial (as all words are winning) — we assume $k \geq 1$. The inequality $k < \frac{1}{\lambda} - 1$ is equivalent to $\frac{1}{\lambda} > \frac{k}{1-\lambda}$. In this case, there are only three possible gaps:

> We simply use that $\frac{1}{\lambda} - 1 = \frac{1-\lambda}{\lambda}$ and then divide by $1 - \lambda$.

- for $w \in 0^*$, $\mathsf{gap}(w) = 0$.
- for $w \in 0^* c$ with $c \geq 1$, then $\frac{\mathsf{DS}_\lambda(w)}{\lambda^{|w|}} = \frac{c}{\lambda} \geq \frac{1}{\lambda} > \frac{k}{1-\lambda} = -\mathsf{minDS}$ — so for any word $w \in 0^* c C^*$, $\mathsf{gap}(w) = \top$.
- for $w \in 0^* c C^*$ with $c \leq -1$, symmetrically, $\mathsf{gap}(w) = \bot$.

These three possible gaps clearly correspond to different equivalence classes for the right congruence $\sim$, so there are three such equivalence classes. Hence, the prefix classifier $\mathcal{S}_W$ has three states $[\varepsilon]_\sim$, $[1]_\sim$, and $[-1]_\sim$.

> The prefix classifier is then
>
>

We now assume that $k \geq \lceil \frac{1}{\lambda} - 1 \rceil$. The left-to-right implication of (5.4) is clear in the cases in which all, or none, of the continuations are winning. The difficult case is when both w_1 and w_2 have a rational gap. We show that if their gaps are different rational numbers, then they have different winning continuations. We assume w.l.o.g. that $\mathsf{gap}(w_2) < \mathsf{gap}(w_1)$. We show that there is an infinite continuation that has a discounted sum exactly equal to $-\mathsf{gap}(w_1)$: this infinite continuation is winning after w_1 but losing after w_2.

Figure 5.15: Prefix classifier of $\mathsf{DS}_\lambda^{\geq 0}$ for $\lambda = \frac{1}{2}$ and $C = \{-2, -1, 0, 1, 2\}$. The value in a state is the gap characterizing the equivalence class for $\sim$ corresponding to that state. Here, $\frac{\sup C}{1-\lambda} = 4$ and $\frac{\inf C}{1-\lambda} = -4$. The asymmetry around 0 comes from the "≥ 0" inequality in the definition of the condition: when state −4 is reached, there is exactly one winning continuation (2^ω), but a state with gap value 4 would only have winning continuations (hence, it is part of state ⊤). Notice that we can build a parity automaton on top of this structure that recognizes $\mathsf{DS}_\lambda^{\geq 0}$: an infinite word is winning as long as it does not reach ⊥.

Showing that there exists $w \in C^\omega$ such that $\mathsf{DS}_\lambda(w) = -\mathsf{gap}(w_1)$ amounts to showing that there is a representation of $-\mathsf{gap}(w_1)$ in the (rational but not necessarily integral) base $\frac{1}{\lambda}$ with digits in C, with one digit before the decimal point. We can adapt the well-known *greedy expansion* [Rén57] to our context to show this (details in Subsection 5.7.2, Proposition 5.7.4 below).

[Rén57]: Rényi (1957), *Representations for real numbers and their ergodic properties*

It is left to show that there are finitely many gap values if and only if λ equals $\frac{1}{n}$ for some integer $n \geq 1$. One implication is clear: if $\lambda = \frac{1}{n}$ for some integer $n \geq 1$, then there are finitely many possible gaps as gaps are then always integers between minDS and maxDS, ⊤, or ⊥. We illustrate this implication by depicting the prefix classifier of $\mathsf{DS}_\lambda^{\geq 0}$ for $\lambda = \frac{1}{2}$ and $k = 2$ in Figure 5.15. The proof of the other implication is provided in Subsection 5.7.2, Proposition 5.7.5. □

Corollary 5.7.3 *Let* $\lambda \in (0, 1) \cap \mathbb{Q}$, $k \in \mathbb{N}$, *and* $C = [-k, k] \cap \mathbb{Z}$.

▶ *If* $k < \frac{1}{\lambda} - 1$, *then* $\mathsf{DS}_\lambda^{\geq 0}$ *is memoryless-determined.*
▶ *If* $k \geq \lceil \frac{1}{\lambda} - 1 \rceil$, *then* $\mathsf{DS}_\lambda^{\geq 0}$ *is chromatic-finite-memory-determined if and only if* λ *is equal to* $\frac{1}{n}$ *for some integer* $n \geq 1$.

This result does not provide optimal memory bounds: when $\mathsf{DS}_\lambda^{\geq 0}$ is ω-regular, it is actually always memoryless-determined. This improved memory bound follows from the fact that under this hypothesis, $\mathsf{DS}_\lambda^{\geq 0}$ is actually a *regular safety objective* and its complement is a *regular reachability objective*, for which memory is characterized in Chapter 7.

Proof. This follows from Propositions 5.7.1 and 5.7.2, thanks to Theorem 5.4.1. The only thing to clarify is that memoryless strategies suffice in case $k < \frac{1}{\lambda} - 1$. The proof of Proposition 5.7.2 tells us that in this case, $\mathsf{DS}_\lambda^{\geq 0}$ is ω-regular and can be recognized by a parity automaton that can be built on top of $\mathcal{S}_W \otimes \mathcal{M}_{\mathsf{triv}}$, which has three states. To use this structure as a memory structure, we can notice that the game is already over in states $[1]_\sim$ and $[-1]_\sim$ (as every continuation wins or every continuation loses). Thus, it is not necessary to use these states to play, and we can consider that we always stay in state $[\varepsilon]_\sim$. □

5.7.2 Missing proofs for the discounted sum application

We prove the two properties used in Proposition 5.7.2 whose proofs were omitted. We use notations from the proof of Proposition 5.7.2 itself.

This subsection contains technical details required for the proof of Proposition 5.7.2, but brings little insight into strategy complexity of games. The uninterested reader may skip ahead to Subsection 5.7.3.

Proposition 5.7.4 *Let* $\lambda \in (0,1) \cap \mathbb{Q}$, $k \in \mathbb{Z}$ *such that* $k \geq \lceil \frac{1}{\lambda} - 1 \rceil$, *and* $C = [-k, k] \cap \mathbb{Z}$. *For any real number* x, $\frac{-k}{1-\lambda} \leq x \leq \frac{k}{1-\lambda}$, *there exists* $w \in C^\omega$ *such that* $x = \mathsf{DS}_\lambda(w)$.

Proof. This problem can be rephrased as a number representation problem: we are looking for a sequence of "digits" $(x_i)_{i \geq 0}$ in C such that $x = x_0.x_1 x_2 \ldots$ in base $\frac{1}{\lambda}$, i.e., such that

$$x = \sum_{i=0}^{\infty} x_i \lambda^i.$$

Notice that $\sum_{i=0}^{\infty} x_i \lambda^i = \mathsf{DS}_\lambda(x_0 x_1 \ldots)$. It is known that every number $x \in [0,1)$ has (at least) one representation $0.x_1 x_2 \ldots$ in base $\frac{1}{\lambda}$ with digits in $\{0, 1, \ldots, \lceil \frac{1}{\lambda} - 1 \rceil\}$, and one such representation can be found using the *greedy expansion* [Rén57].

[Rén57]: Rényi (1957), *Representations for real numbers and their ergodic properties*

We adapt this greedy expansion to our setting (for a potentially greater x and larger set C of digits). Let $x \in \mathbb{R}$ such that $\frac{-k}{1-\lambda} \leq x \leq \frac{k}{1-\lambda}$. We deal with the case $x \geq 0$ — the negative case is symmetric. We set $x_0 = \min(k, \lfloor x \rfloor)$; clearly, $x_0 \leq x$. Then inductively, if $x_0, \ldots, x_{l-1}$ have been defined, we define x_l as the greatest integer in $\{0, \ldots, k\}$ such that

$$\sum_{i=0}^{l} x_i \lambda^i \leq x.$$

The series $\sum_{i=0}^{\infty} x_i \lambda^i$ is converging, as every term is non-negative and it is bounded from above by x. We show that it converges to x, which ends the proof. Let $\epsilon \geq 0$. Assume by contradiction that $\sum_{i=0}^{\infty} x_i \lambda^i \leq x - \epsilon$. Let j be the least index such that $\lambda^j \leq \epsilon$. Clearly, for any $j' \geq j$, $x_{j'} = k$ — otherwise, a greater digit could have been picked during the inductive greedy selection. Still, not every digit $x_0, x_1, \ldots$ can be k, as $\sum_{i=0}^{\infty} k\lambda^i = \frac{k}{1-\lambda} > x - \epsilon$. Let l be the greatest index such that $x_l \neq k$. We show that a digit $\geq x_l + 1$ should have been picked instead of x_l for the digit at index l, leading to a contradiction. To do so, it is sufficient to show that

$$(x_l + 1)\lambda^l + \sum_{i=0}^{l-1} x_i \lambda^i \leq x.$$

We have

$$(x_l + 1)\lambda^l + \sum_{i=0}^{l-1} x_i \lambda^i = \sum_{i=0}^{\infty} x_i \lambda^i + \lambda^l - \sum_{i=l+1}^{\infty} x_i \lambda^i$$

$$= \sum_{i=0}^{\infty} x_i \lambda^i + \lambda^l - \sum_{i=l+1}^{\infty} k\lambda^i \quad \text{as } x_i = k \text{ for } i \geq l+1$$

$$= \sum_{i=0}^{\infty} x_i \lambda^i + \lambda^l - \frac{k\lambda^{l+1}}{1-\lambda}$$

$$= \sum_{i=0}^{\infty} x_i \lambda^i + \lambda^l (1 - \frac{k\lambda}{1-\lambda}).$$

Since $\frac{k\lambda}{1-\lambda} \geq \frac{\lceil \frac{1}{\lambda} - 1\rceil \lambda}{1-\lambda} = \lceil \frac{1-\lambda}{\lambda}\rceil \frac{\lambda}{1-\lambda} \geq 1$, we have that $\lambda^l(1 - \frac{k\lambda}{1-\lambda}) \leq 0$, which implies that

$$(x_l + 1)\lambda^l + \sum_{i=0}^{l-1} x_i\lambda^i \leq \sum_{i=0}^{\infty} x_i\lambda^i < x,$$

a contradiction. We conclude that $x = \mathsf{DS}_\lambda(x_0 x_1 \ldots)$. $\qquad\square$

Proposition 5.7.5 *Let $\lambda \in (0,1) \cap \mathbb{Q}$, $k \in \mathbb{Z}$ such that $k \geq \lceil \frac{1}{\lambda} - 1\rceil$, and $C = [-k, k] \cap \mathbb{Z}$. If $\lambda \neq \frac{1}{n}$ for all integers $n \geq 1$, the* gap *function takes infinitely many values.*

Proof. We assume that $\lambda = \frac{p}{q}$ with $p, q \in \mathbb{N}$ co-prime, $p \geq 2$ and $q > p$, and we show that the gap function takes infinitely many values. To do so, we exhibit an infinite word $w = c_1 c_2 \ldots \in C^\omega$ such that the sequence of rationals $(\mathsf{gap}(c_1 \ldots c_i))_{i \geq 1}$ never takes the same value twice.

We will use the following inductive property of gaps: for $w \in C^*$ and $c \in C$,

$$\mathsf{gap}(wc) = \frac{\mathsf{gap}(w)}{\lambda} + \frac{\lambda^{|w|-1}c}{\lambda^{|w|}} = \frac{1}{\lambda}\left(\mathsf{gap}(w) + c\right), \qquad (5.5)$$

unless some gap in this equation equals $\top$ or $\bot$. Notice that under our hypotheses, $\lceil \frac{1}{\lambda} - 1\rceil = \lfloor \frac{1}{\lambda}\rfloor$ (this equality does not hold when $\lambda = \frac{1}{n}$ for some integer $n \geq 1$).

We set $c_1 = 1$. Then, $\mathsf{gap}(c_1) = \frac{1}{\lambda} = \frac{q}{p}$. Inductively, if $c_1, \ldots, c_{i-1}$ are defined, we set $c_i = -\lfloor \mathsf{gap}(c_1 \ldots c_{i-1})\rfloor$ (we remove the largest possible integer from the current gap, while keeping a positive gap value). We set $g_i = \mathsf{gap}(c_1 \ldots c_i)$ for conciseness.

We first show that if all g_i's are rational (i.e., are not $\top$ or $\bot$), then no two g_i's can be equal. To do so, we show inductively that the reduced denominator of fraction g_i is p^i for all $i \geq 1$. This is true for $i = 1$. For $i > 1$, assume it is true for $i - 1$. Then, $g_{i-1} = \frac{m}{p^{i-1}}$ for some m co-prime with p. Using Equation (5.5),

$$g_i = \frac{1}{\lambda}\cdot\left(g_{i-1} + c_i\right) = \frac{1}{\lambda}\cdot\left(\frac{m}{p^{i-1}} + c_i\right) = \frac{q(m + c_i p^{i-1})}{p^i}.$$

This last fraction is irreducible: q and p are co-prime, and the fact that m and p are co-prime implies that $m + c_i p^{i-1}$ and p are co-prime.

We now prove by induction that our scheme is well-defined by showing that for all $i \geq 1$, $c_i \in C$ and $0 < g_i \leq \frac{1}{\lambda}$. This is true for $i = 1$ (as $k \geq 1$ for any possible value of λ). For $i > 1$, if this is true for $i - 1$,
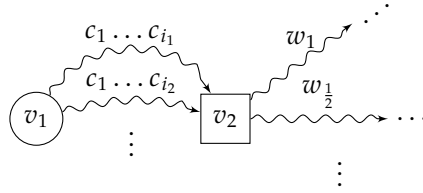
then $-c_i = \lfloor g_{i-1} \rfloor \leq \lfloor \frac{1}{\lambda} \rfloor$, so $c_i \in C$. Moreover, $g_i = \frac{1}{\lambda} (g_{i-1} + c_i)$. Since g_i's cannot be integers (as their reduced denominator is not 1 by an earlier property), we have that $g_{i-1} + c_i$ is not an integer either. Therefore, $0 < g_{i-1} + c_i < 1$, so $0 < g_i < \frac{1}{\lambda}$. As $\frac{1}{\lambda} < \frac{k}{1-\lambda} = \mathsf{maxDS}$, the values of the considered gaps are never $\top$ or $\bot$. $\qquad\square$

Remark 5.7.6 For $\lambda \in (0, 1) \cap \mathbb{Q}$ with $\lambda \neq \frac{1}{n}$ for all $n \geq 1$, $k \geq \lceil \frac{1}{\lambda} - 1 \rceil$, and $C = [-k, k] \cap \mathbb{Z}$, Proposition 5.7.2 along with Theorem 5.4.1 implies that any memory structure is insufficient to play optimally (for at least one player). However, this does not directly give an explicit arena in which some player requires infinite memory to play optimally. Here, we show how to construct such an arena given the extra results from Subsection 5.7.2.

The proof of Proposition 5.7.5 gives us $c_1 c_2 \ldots \in C^\omega$ such that $(\mathsf{gap}(c_1 \ldots c_i))_{i \geq 1}$ is a sequence of distinct values in $[0, \mathsf{maxDS}]$. Hence, by compactness of $[0, \mathsf{maxDS}]$, there exists a subsequence $(i_j)_{j \geq 1}$ and $x \in [0, \mathsf{maxDS}]$ such that $\lim_{j \to \infty} \mathsf{gap}(c_1 \ldots c_{i_j}) = x$. We can moreover extract a subsequence such that either all elements are greater than x, or all elements are less than x. We assume w.l.o.g. that for all $j \geq 1$, $\mathsf{gap}(c_1 \ldots c_{i_j}) < x$ (this implies that $x \neq 0$). The proof is symmetric if all the gaps are greater than x (which would imply that $x \neq \mathsf{maxDS}$).

By Proposition 5.7.4, for all $\epsilon > 0$ sufficiently small, there exists $w_\epsilon \in C^\omega$ such that $\mathsf{DS}_\lambda(w_\epsilon) = -x + \epsilon$. We can define an infinite arena in which $\mathcal{P}_2$ needs infinite memory to win, depicted in Figure 5.16. In this arena, $\mathcal{P}_1$ may choose to reach a gap arbitrarily close (but not equal) to x in v_2, and then $\mathcal{P}_2$ is always able to bring the discounted sum below 0 by choosing a word reaching a discounted sum sufficiently close to $-x$.

5.7.3 Other objectives

Mean payoff. Let $C \subseteq \mathbb{Q}$ be non-empty. We consider the mean-payoff objective $\mathsf{MP}^{\geq 0}$, containing the infinite words whose mean payoff is non-negative. This condition is $\mathcal{M}_{\mathsf{triv}}$-prefix-independent for any set of colors. However, we also know that infinite memory is necessary to play optimally in some infinite arenas (this was discussed in Section 3.2). Here, we show that chromatic-finite-memory strategies do not suffice to play optimally (even when $C = \{-1, 1\}$) by analyzing cycle-consistency of $\mathsf{MP}^{\geq 0}$. If we

consider, for $n \in \mathbb{N}$,

$$w_n = \underbrace{1, \dots, 1}_{n \text{ times}}, \underbrace{-1, \dots, -1}_{n+1 \text{ times}},$$

we have that $(w_n)^\omega$ is losing for all $n \geq 0$, but the infinite word $w_0 w_1 w_2 \dots$ has a mean payoff of 0 and is thus winning. This shows directly that $\mathsf{MP}^{\geq 0}$ is not $\mathcal{M}_{\mathsf{triv}}$-cycle-consistent. The argument can be adapted to show that $\mathsf{MP}^{\geq 0}$ is not $\mathcal{M}$-cycle-consistent for any structure $\mathcal{M}$.

Lemma 5.7.7 *For all structures $\mathcal{M}$, $\mathsf{MP}^{\geq 0}$ is not $\mathcal{M}$-cycle-consistent.*

Proof. Let $\mathcal{M} = (M, m_{\mathsf{init}}, \alpha_{\mathsf{upd}})$ be a structure. For $n \in \mathbb{N}$, let

$$w_n = \underbrace{1, \dots, 1}_{n \text{ times}}, \underbrace{-1, \dots, -1}_{n+1 \text{ times}}.$$

Let M_n be the set of states $m \in M$ such that there exists $k \geq 1$ with $m = \alpha^*_{\mathsf{upd}}(m, w_n^k)$. Each M_n is non-empty as, M being finite, iterating the function $m \mapsto \alpha^*_{\mathsf{upd}}(m, w_n)$ necessarily goes multiple times through at least one state. Let $m \in M$ be a state in set M_n for infinitely many n's, and w be any finite word in $\mathsf{col}^*(\Pi_{m_{\mathsf{init}}, m})$. Let $n_1, n_2, \dots$ be the indices such that $m \in M_{n_i}$, and let $k_1, k_2, \dots$ be such that $m = \alpha^*_{\mathsf{upd}}(m, w_{n_i}^{k_i})$. Every word $w_{n_i}^{k_i}$ is a losing cycle after any finite word (in particular after w). However, it is possible to find a subsequence of $(w_{n_i}^{k_i})_{i \geq 1}$ with a non-negative mean payoff by always taking a word w_{n_i} that bring the sum of the colors above 0 during the first n_i 1's. $\square$

Total payoff. Let $C \subseteq \mathbb{Q}$ be non-empty. We define the *total-payoff function* $\mathsf{TP} \colon C^\omega \to \mathbb{R} \cup \{-\infty, \infty\}$ such that for $w = c_1 c_2 \dots \in C^\omega$,

$$\mathsf{TP}(w) = \limsup_{n \to \infty} \sum_{i=1}^{n} c_i.$$

We define the objective $\mathsf{TP}^{\geq 0} = \{w \in C^\omega \mid \mathsf{TP}(w) \geq 0\}$ as the set of infinite words whose total payoff is non-negative.

The right congruence $\sim$ of $\mathsf{TP}^{\geq 0}$ does not have finite index, even for $C = \{-1, 1\}$: we indeed have that $-1 > (-1, -1) > \dots$ is an infinite decreasing sequence of prefixes. Condition $\mathsf{TP}^{\geq 0}$ is therefore not $\mathcal{M}$-prefix-independent for any structure $\mathcal{M}$. We can also show that $\mathsf{TP}^{\geq 0}$ is not $\mathcal{M}$-cycle-consistent for any $\mathcal{M}$, using the exact same argument as for $\mathsf{MP}^{\geq 0}$. Chromatic-finite-memory strategies are therefore insufficient to play optimally for $\mathsf{TP}^{\geq 0}$ in infinite arenas. Once again, this situation contrasts with the case of finite arenas, in which memoryless strategies suffice to play optimally [GZ04].

[GZ04]: Gimbert et al. (2004), *When Can You Play Positionally?*

5.8 Wrap-up

We proved an equivalence between chromatic finite-memory determinacy of an objective in games on infinite graphs and ω-regularity of the corresponding language of infinite words, generalizing a result by Colcombet and Niwiński [CN06]. A "strategic" consequence of our result is that chromatic finite-memory determinacy restricted to the one-player games of both players implies the seemingly stronger chromatic finite-memory determinacy over two-player zero-sum games. A "language-theoretic" consequence is a relation between the representation of ω-regular languages by parity automata and the memory structures used to play optimally in zero-sum games, using as a tool the prefix classifier classifying the equivalence classes for the right congruence.

One possible improvement over our result is to deduce tighter chromatic memory requirements in two-player games compared to one-player games: our proof technique gives as an upper bound on the two-player memory requirements a product between the prefix classifier and a sufficient structure for one-player arenas, but smaller structures often suffice (Conjecture 5.4.5). This behavior contrasts with the case of finite arenas, in which a memory structure sufficient for both players in finite one-player arenas also suffices in finite two-player arenas (Chapter 4). More generally, our results do not provide precise (chromatic) memory requirements of ω-regular objectives, for which there are existing results on Muller conditions [DJW97; Cas22], and to which the second part *Obtaining precise memory requirements* of this thesis is devoted.

[CN06]: Colcombet et al. (2006), *On the positional determinacy of edge-labeled games*

[DJW97]: Dziembowski et al. (1997), *How Much Memory is Needed to Win Infinite Games?*
[Cas22]: Casares (2022), *On the Minimisation of Transition-Based Rabin Automata and the Chromatic Memory Requirements of Muller Conditions*

OBTAINING PRECISE MEMORY REQUIREMENTS

Known and unknown memory requirements of ω-regular objectives $\Big|$ 6

Our focus in the rest of this second part *Obtaining precise memory requirements* is to obtain, understand, and compute precise memory requirements for interesting classes of objectives. Due to their relevance in logic and synthesis, as well as their deep link with finite-memory determinacy (see Chapter 5), focusing on *ω-regular objectives* is a natural way forward. This is also hopefully a stepping stone to study the strategy complexity of other more involved objectives. We sketch here a picture of what is known and unknown about the memory requirements of ω-regular objectives, and what answers we bring in the subsequent chapters. We look at this question through the viewpoint of the contributions in Chapters 4 and 5, highlighting what is still missing to get a complete picture.

This chapter serves as a motivation and an introduction to Chapters 7 and 8, which contain our main contributions to this topic.

6.1 The missing pieces

As was stated in Theorems 5.4.1 and 5.4.3, one can relate the memory requirements of an ω-regular objective to its representation as a deterministic parity automaton (DPA). We may wonder how close this result brings us to characterizing precisely the memory requirements of ω-regular objectives. Albeit quite general, there are still multiple questions about memory requirements that this result fails to answer precisely. We introduce these questions by highlighting two of its limitations.

1. The first limitation comes from the asymmetry of its two implications. In one direction, which is the novel contribution from Theorem 5.4.3, we start from a sufficient memory structure $\mathcal{M}$ for some objective W and show that the objective can then be represented as a DPA built on top of the automatic structure $\mathcal{M} \otimes \mathcal{S}_W$. In the other direction, we simply use the known memoryless determinacy of parity conditions: ω-regular objectives are recognized by a DPA, and keeping track of the information given by this DPA transforms any game into a (larger) game endowed with a (simpler) parity condition. Now, what does this tell us about the memory requirements of an ω-regular objective? We know two things:

   ▸ a minimal memory structure always has at most as many states as any DPA representing the objective;

▶ a minimal memory structure and a minimal DPA differ at most by an "$\mathcal{S}_W$ factor".

Yet, we do not know in general how to get the minimal memory requirements of an ω-regular objective from its representation as a DPA. We get close to a satisfying answer for prefix-independent ω-regular objectives W, for which $\mathcal{S}_W = \mathcal{M}_{\text{triv}}$ has just one state: a minimal memory structure (sufficient for *both* players) is then exactly a minimal DPA recognizing the objective (this follows from Theorem 5.4.1). But even then, this does not give the full picture, which we now argue.

2. The second, perhaps more fundamental limitation of Theorem 5.4.3 is that it makes an assumption on the memory requirements of *both* players simultaneously, as it asks for a memory structure sufficient for both players. This assumption therefore conceals a possibly large gap between the individual memory requirements of the two players. This limitation also applied to Theorem 4.4.4 (which deals with games played on finite arenas), which cannot be used in general to give tight bounds about memory requirements of each individual player in two-player games played on finite arenas.

We illustrate these two limitations in a small example: in this example, the representation of an objective as a DPA is an upper (but not a tight) bound on the memory requirements, and the memory requirements of each player differ.

Example 6.1.1 Let $C = \{a, b\}$. We consider the objective $W = \text{Büchi}(a) \cup (C^* aa C^\omega)$ of words that see a infinitely often or see a twice in a row at some point. This objective is recognized by the DBA with three states depicted in Figure 6.1 (left), and it is not possible to recognize it using a DBA (or even with a DPA) with fewer states. We therefore know that both players can play optimally using three states of memory, using the memory structure underlying this automaton. The prefix classifier $\mathcal{S}_W$ of this objective has three states corresponding to three equivalence classes of finite words. Its structure corresponds to the underlying structure of the DBA in Figure 6.1. According to Theorem 5.4.3, this suggests that $\mathcal{P}_1$ and $\mathcal{P}_2$ may need between one and three states of memory to play optimally.

It turns out that $\mathcal{P}_1$ can play optimally with just one state of memory in all arenas (i.e., W is half-positional), which can be proved using upcoming results from Chapter 8. Meanwhile, $\mathcal{P}_2$ cannot play optimally with just one state of memory, as is witnessed by the arena in Figure 6.1 (right). If the play starts in v_1, we observe that $\mathcal{P}_2$ loses by not using the loop on v_2 and going immediately to v_3, as well as by staying infinitely often in v_2. Player $\mathcal{P}_2$ can actually win, but needs to loop at least once (and finitely many times) on v_2 before going to v_3, which cannot be done without memory. For this objective, $\mathcal{P}_2$ can play optimally with

We recall that a DBA is a special kind of DPA with only priorities 1 and 2 (Remark 2.7.9), so we can use a DBA in the results of Chapter 5.

The reason we cannot represent it with fewer states is that we need at least one state per equivalence class of the right congruence (Lemma 2.8.8), of which W has three.

See Example 8.3.3 for a proof and a more thorough discussion of this example.
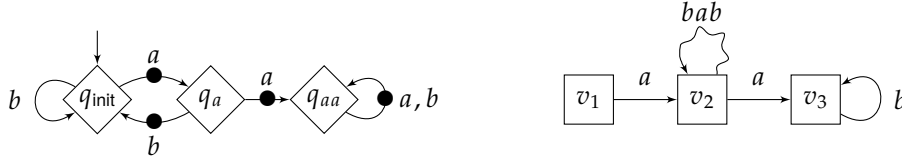
Figure 6.1: A DBA representing the objective $W = \text{Büchi}(a) \cup (C^* aa C^\omega)$ from Example 6.1.1 (left), and an arena in which $\mathcal{P}_2$ cannot play optimally with a memoryless strategy (right).

> two states of memory: intuitively, $\mathcal{P}_2$ must keep track of whether the current history is in q_{init} or q_a, but there is no point in keeping track of state q_{aa}, as the play is already lost in that state for $\mathcal{P}_2$.

The following questions therefore remain: given an ω-regular objective, what is a *minimal* (i.e., with as few states as possible) memory structure sufficient to play optimally for both players? And for a *single* player? And how to compute it? A less ambitious (but still open) question would be to understand the memoryless case: how to characterize and decide memoryless determinacy or half-positionality? And would these precise results give us even more information on the representation of ω-regular objectives, perhaps using other acceptance conditions than the parity one? Progress toward these questions can be seen as strengthenings of Theorem 5.4.3.

In the next section, we give a brief overview of *Muller conditions*, for which there already are complete answers to these questions thanks to existing works published between 1982 and 2022.

6.2 The case of Muller conditions

Muller conditions (Definition 2.4.11) are objectives whose winning words depend solely on the set of colors seen *infinitely often*. They are usually specified by a set $\mathcal{F} \subseteq 2^C$ of sets of colors. The derived Muller condition then contains the set of words $w \in C^\omega$ such that the set of colors seen infinitely often by w is a set of $\mathcal{F}$. They are in particular prefix-independent, i.e., their prefix classifier has just one state. A systematic study of the memory requirements of Muller conditions started in the 1980s, with first general upper bounds through the *later appearance record construction* [GH82; McN93], culminating in a complete characterization of their (chaotic) memory requirements for each individual player [DJW97].

Chromatic versus chaotic memory. About *chromatic* memory requirements of Muller conditions, we mention a recent work by Casares [Cas22] that characterizes the chromatic memory requirements of Muller conditions for each individual player, once again through their representations using a specific type of automaton. It uses the *Rabin* acceptance condition, which subsumes parity acceptance conditions.

[GH82]: Gurevich et al. (1982), *Trees, Automata, and Games*

[McN93]: McNaughton (1993), *Infinite Games Played on Finite Graphs*

[DJW97]: Dziembowski et al. (1997), *How Much Memory is Needed to Win Infinite Games?*

[Cas22]: Casares (2022), *On the Minimisation of Transition-Based Rabin Automata and the Chromatic Memory Requirements of Muller Conditions*

> **Theorem 6.2.1** ([Cas22, Theorem 27]) *Let $W \subseteq C^\omega$ be a Muller condition and $\mathcal{M}$ be a memory structure. Structure $\mathcal{M}$ suffices for $\mathcal{P}_1$ for objective W if and only if W is recognized by a deterministic Rabin automaton built on top of $\mathcal{M}$.*

[Cas22]: Casares (2022), *On the Minimisation of Transition-Based Rabin Automata and the Chromatic Memory Requirements of Muller Conditions*

Once again, one direction was already known: Rabin conditions have been known to be half-positional for some time [Kla94] — but unlike parity conditions, their complement (*Streett conditions*) may not be. The other, novel direction of Theorem 6.2.1 was obtained thanks to results about the representation of ω-regular objectives [CCF21].

[Kla94]: Klarlund (1994), *Progress Measures, Immediate Determinacy, and a Subset Construction for Tree Automata*

[CCF21]: Casares et al. (2021), *Optimal Transformations of Games and Automata Using Muller Conditions*

This provides, in the special case of Muller conditions, a characterization of the memory requirements of each player without any blow-up in any direction of the equivalence, and independently of the memory requirements of the other player. It therefore goes beyond the two limitations of Theorem 5.4.3 sketched in Subsection 6.1. As a bonus, this result allows linking the problem of finding a minimal memory structure to the problem of minimizing a Rabin automaton recognizing a Muller condition, and implies that the related decision problem (given a Muller condition, is there a sufficient memory structure with $\leq k$ states for a fixed k?) is NP-complete.

The characterization for chaotic memory [DJW97] was also recently revisited by Casares, Colcombet, and Lehtinen [CCL22], giving once again a link between memory requirements and representation of the objectives through automata.

[DJW97]: Dziembowski et al. (1997), *How Much Memory is Needed to Win Infinite Games?*

[CCL22]: Casares et al. (2022), *On the Size of Good-For-Games Rabin Automata and Its Link with the Memory in Muller Games*

> **Theorem 6.2.2** ([CCL22, Theorem 5]) *Let $W \subseteq C^\omega$ be a Muller condition. Player $\mathcal{P}_1$ can play optimally in all arenas with a chaotic memory structure with k states if and only if W is recognized by a* good-for-games *Rabin automaton with k states.*

Good-for-games automata are a class of non-deterministic automata whose non-determinism can be resolved with a strategy that only looks at the past. They were first introduced by Henzinger and Piterman [HP06] to simplify the synthesis process by allowing some constrained non-determinism in the automata instead of requiring deterministic automata. The result of Theorem 6.2.2 shows a way to express even *chaotic memory structures* using representations of ω-regular objectives. This is a promising research direction that we leave as a future prospect.

[HP06]: Henzinger et al. (2006), *Solving Games Without Determinization*

Why does it not solve the problem?   The work [DJW97] is sometimes quoted as fully solving the problem of the memory requirements of ω-regular objectives. For example, in [BCJ18], we can read

[BCJ18]: Bloem et al. (2018), *Graph Games and Reactive Synthesis*

> *The results of Dziembowski et al. [DJW97] give precise memory requirements for strategies in 2-player games with ω-regular objectives.*

We argue that this is only true for the class of Muller conditions, which is a subclass of the prefix-independent ω-regular objectives, and that it only gives an *upper bound* on the memory requirements of ω-regular objectives in general.

An ω-regular objective cannot usually be expressed directly as a Muller condition, but as a Muller condition *on top of an automaton structure* (i.e., a deterministic Muller automaton). Let us look back at objective $W = \mathsf{B\ddot{u}chi}(a) \cup (C^*aaC^\omega)$ from Example 6.1.1 above. This objective cannot be written as a Muller condition on C (as "finite behavior" matters). If we add to our game the information from the DBA on the left of Figure 6.1 (i.e., by taking the product of the arena with the DBA), then we can indeed reduce our problem to a Muller condition thanks to the additional automaton structure. In this augmented game, by adding information from the automaton to the colors, objective W can be realized as Muller condition $\mathsf{Muller}(\{F \in 2^Q \mid q_a \in F \vee q_{aa} \in F\})$. We can then apply the results from [DJW97] to this Muller condition and find that $\mathcal{P}_1$ needs just one state of memory to play optimally.

[DJW97]: Dziembowski et al. (1997), *How Much Memory is Needed to Win Infinite Games?*

Nevertheless, the sufficiency of this one state of memory comes after taking the product of arenas with the DBA, which essentially consists in playing with three memory states! This therefore provides an upper bound on the memory requirements: it suffices to play with $3 \cdot 1 = 3$ memory states. This upper bound is not tight, as discussed in Example 6.1.1: $\mathcal{P}_1$ can actually play optimally without any memory, not even the information given by the structure of the DBA.

Goals. We seek here an extension of the results about Muller conditions to more ω-regular objectives, focusing on chromatic memory requirements. A generalization keeping the same formulation as Theorems 6.2.1 and 6.2.2 is hopeless: for some objectives, the memory is smaller than the smallest deterministic or good-for-games (or even non-deterministic) Rabin automaton recognizing the objective (see, e.g., Example 6.1.1 or objective W_2 in Example 5.4.2). In what follows, we discuss two important classes of ω-regular objectives, orthogonal to Muller conditions due to their heavy reliance on the automatic structure rather than on the "infinite behavior".

▶ In Chapter 7, we characterize the chromatic memory requirements of *regular reachability objectives*, which are objectives derived from regular languages of *finite* words. For such objectives, $\mathcal{P}_1$ simply wants that a prefix of the infinite play realizes a word from a fixed regular language, while $\mathcal{P}_2$ (who plays for a *regular safety objective*) wants to avoid this for an infinite duration. Despite the simplicity of these objectives, there remained many gaps in the understanding of their memory requirements.

Notice the distinction between *regular* and *ω-regular*.

▶ In Chapter 8, we focus on a more general class of ω-regular objectives, which are those recognized by a *deterministic Büchi automaton*. We

do not get a full picture of their memory requirements, but we characterize those that are *half-positional*, i.e., for which memoryless strategies suffice to implement optimal strategies for $\mathcal{P}_1$.

In both cases, we also study the computational problem of deciding whether small memory structures suffice to play optimally and, for regular objectives, of synthesizing them.

The case of regular languages | 7

We focus in this chapter on some of the simplest possible objectives: those that can be expressed with regular languages of *finite* words — thus, with deterministic finite automata. For such an objective, the goal of $\mathcal{P}_1$ is that, eventually, a sequence of colors along the play belongs to a fixed regular language. Although these games are easy to solve, determining minimal memory structures to play optimally is not trivial. We obtain characterizations of the chromatic memory requirements for such objectives for both players, from which we derive complexity-theoretic statements: deciding whether there exist small memory structures sufficient to play optimally is NP-complete for both players. We also discuss a slight generalization of these objectives to topologically open and topologically closed objectives, which can be seen as objectives expressible using deterministic *infinite* automata.

This chapter is also an opportunity to apply the one-to-two-player lift from Chapter 4 to a concrete class of objectives and to shed new light on the $\mathcal{M}$-monotony notion.

The contributions from this chapter are based on joint work with Patricia Bouyer (Université Paris-Saclay, CNRS, ENS Paris-Saclay, LMF), Nathanaël Fijalkow (CNRS, LaBRI, Université de Bordeaux & University of Warsaw), and Mickael Randour (F.R.S.-FNRS & Université de Mons) currently available on arXiv [BFRV22]. We thank Antonio Casares and Igor Walukiewicz for discussions that were essential in starting this work.

7.1 Motivation

We have seen in Chapter 6 that the memory requirements of ω-regular objectives are not entirely understood. Perhaps one of the most pressing open questions in that direction is the case of *regular objectives*, meaning the ω-regular objectives concerned with finite duration. In this setting, an objective is induced by a regular language over finite words, and the goal of the first player is that, eventually, the sequence of colors along the play belongs to this language. We call such an objective a *regular reachability objective*. The opponent's objective is then to ensure that the sequence of colors *never* belongs to the language, which describes a *regular safety objective*.

Contributions.   We study the chromatic memory requirements of both regular reachability and regular safety objectives. For both cases, we give a combinatorial characterization of the memory structures sufficient to play optimally in all arenas (of any cardinality). As a by-product of the characterization, we obtain complexity-theoretic statements: given as an input a deterministic finite automaton representing the objective,

- ▶ deciding whether a fixed memory structure suffices to play optimally in all arenas can be done in polynomial time;
- ▶ deciding the existence of a sufficient memory structure with a given number of states is NP-complete.

From our characterizations, it also follows that for both regular reachability and regular safety objectives, chromatic and chaotic memory requirements do not coincide.

We also discuss when relevant the extension of our results to the more general classes of *topologically open* and *topologically closed* objectives, which include respectively the regular reachability and regular safety objectives. For consistency, we respectively call these more general objectives *general reachability objectives* and *general safety objectives*.

As a corollary, we obtain a *finite-to-infinite* and an *asymmetric one-to-two-player lift* (Theorem 7.4.14): in order to find the memory requirements of regular objectives for a player over arbitrary — possibly two-player and infinite — game graphs, it suffices to find the memory requirements over *finite one-player* graphs, where this player controls all the vertices.

Additional related works.   The work most related to the present chapter is by Colcombet, Fijalkow, and Horn [CFH14; CFH22], and gives a characterization of *chaotic* memory requirements for general safety objectives. Their constructions strongly rely on the model of chaotic memory; indeed, as a corollary of our results, we will see that already for regular safety objectives, chromatic and chaotic memory requirements do not coincide.

[CFH14]: Colcombet et al. (2014), *Playing Safe*
[CFH22]: Colcombet et al. (2022), *Playing Safe, Ten Years Later*

Regular objectives are also mentioned in [LPR18], where the existence of finite-memory optimal strategies is shown for Boolean combinations of objectives involving regular objectives.

Implementation.   For experimentation purposes, we have implemented algorithms that automatically build a memory structure with a minimal number of states, for both regular reachability and regular safety objectives. Our implementation uses SAT solvers provided by the Python package PySAT [IMM18].

Chapter structure.   We introduce general/regular reachability/safety objectives as well as a few concepts to manipulate them in Section 7.2. We then describe our three kinds of contributions in the subsequent sections: Section 7.3 discusses general and regular safety objectives, Section 7.4 discusses general and regular reachability objectives, and Section 7.5 discusses the computational complexity of finding small memory structures. To ease the reading, some of the more complex proofs are deferred to Section 7.6. We then briefly describe the code implementation in Section 7.7.

7.2 Preliminaries: reachability and safety objectives

General/regular, reachability/safety objectives.   We focus on two types of objectives, both derived from a set $A \subseteq C^*$.

> **Definition 7.2.1** *Let $A \subseteq C^*$.*
> *The* general reachability objective *derived from A, which we denote* **GenReach**(A), *is the objective $\bigcup_{w \in A} wC^\omega$ of infinite words that have (at least) one finite prefix in A.*
> *The* general safety objective *derived from A, which we denote* **GenSafe**(A), *is the objective $\overline{\bigcup_{w \in A} wC^\omega}$ of infinite words that have no finite prefix in A. We have* **GenSafe**$(A) = \overline{\textbf{GenReach}(A)}$.

General reachability and safety objectives are respectively the *topologically open* and *topologically closed sets*, at the first level of the Borel hierarchy. When A is a regular language, we call GenReach(A) a *regular reachability objective* and GenSafe(A) a *regular safety objective*. We then call an objective *regular* if it is a regular reachability or a regular safety objective. Our characterizations apply to regular reachability and safety objectives, but we sometimes discuss when we may generalize our results to the general reachability and safety objectives (i.e., topologically open and closed sets). For computational complexity questions (Section 7.5), we will restrict our focus to *regular* reachability and safety objectives so that an objective can be finitely represented as a DFA. The objectives that we consider are therefore very simple both in terms of their algebraic representation (using

automata representing languages of finite words) and in terms of their topology (they are at the first level of the Borel hierarchy).

Prefix-classifier *automaton*.   Starting from a regular reachability or safety objective $W \subseteq C^\omega$ derived from a set $A \in C^*$, we can associate as usual the prefix classifier $\mathcal{S}_W$ with W. Here, it makes sense to generalize the definition of a prefix classifier to being a deterministic *automaton* rather than just an automaton *structure*, as there is a natural choice for a set of final states.

Prefix classifiers were defined in Definition 2.8.11.

> **Definition 7.2.2** *Let $W \subseteq C^\omega$ be a general safety or reachability objective. The* prefix-classifier automaton *of W is the automaton $\mathcal{D}_W = (Q, C, q_{\text{init}}, \delta, F)$, where Q, q_{init}, and δ are defined as for prefix classifiers, and $F = \{q_{\text{fin}}\}$ where $q_{\text{fin}} = [w]_\sim$ for some $w \in A$ (the choice of w in A does not matter).*

Notice that the final state of such an automaton is always unique and absorbing, i.e., for all $c \in C$, $\delta(q_{\text{fin}}, c) = q_{\text{fin}}$. This matches the intuition that once a word of A is seen and the reachability (resp. safety) game is won (resp. lost), it stays that way for the rest of the game.

One uncommon fact about this definition of automaton $\mathcal{D}_W$ with respect to our usual conventions is that it may have an *infinite* state space. We allow this convenient abuse for objects $\mathcal{D}_W$ throughout this chapter, and we carefully remind when it is allowed to be infinite. Using the well-known Myhill-Nerode theorem [Ner58], we obtain that a general reachability or safety objective W is regular if and only if $\sim$ has finitely many equivalence classes if and only if $\mathcal{D}_W$ is finite. Situations in which $\mathcal{D}_W$ is allowed to be infinite then correspond to instances in which we intend to deal with general (and not only regular) reachability and safety objectives.

[Ner58]: Nerode (1958), *Linear Automaton Transformations*

We have that a general reachability (resp. safety) objective W is equal to $\mathsf{GenReach}(\mathcal{L}(\mathcal{D}_W))$ (resp. to $\mathsf{GenSafe}(\mathcal{L}(\mathcal{D}_W))$) — in examples, we will sometimes start from an automaton to generate an objective. Therefore, all general reachability and safety objectives can be generated from a prefix-classifier automaton. This implies in particular that all automata in this chapter can be assumed to have a single, absorbing final state. We observe that starting from an automaton $\mathcal{D}$, the automaton $\mathcal{D}_{\mathsf{GenReach}(\mathcal{L}(\mathcal{D}))}$ (resp. $\mathcal{D}_{\mathsf{GenSafe}(\mathcal{L}(\mathcal{D}))}$) may not recognize exactly the same regular language of finite words as $\mathcal{D}$, but both automata induce the same general reachability (resp. safety) objective. We limit the illustration of this phenomenon to an example, as it bears no significance to our results.

> **Example 7.2.3** We consider the regular language $L = (a^+ + b)a$, recognized by the DFA in Figure 7.1 (left). It requires six states to be represented as a DFA, including two final states. However, the prefix-classifier automaton (right) of its induced reachability and safety objectives needs just four states, with a single absorbing final state. Intuitively, the two
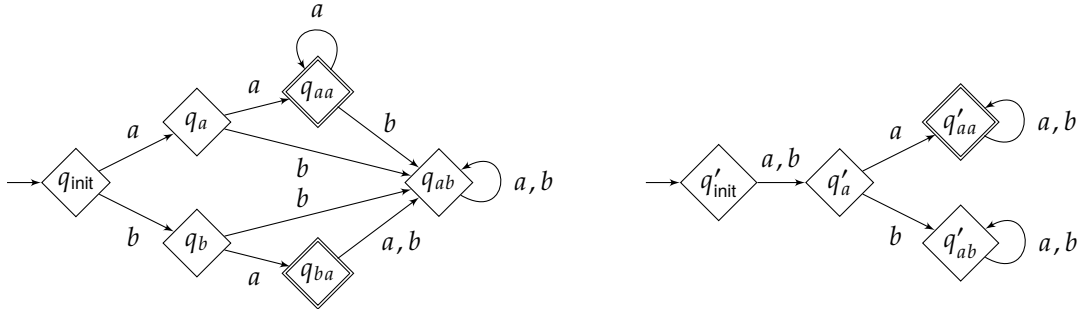
Figure 7.1: DFA $\mathcal{D}$ recognizing $L = (a^+ + b)a$ (left), and the smaller prefix-classifier automaton $\mathcal{D}_W$ of objective $W =$ GenReach(L) (right) (objective GenSafe(L) has the same prefix-classifier automaton). We recall that final states are represented with a double border.

> final states q_{aa} and q_{ba} can be merged into a single final absorbing state q'_{aa}. We can then notice that q_a and q_b have the same transitions and can be merged into q'_a. The regular language it recognizes is CaC^*.

We extend two notations from Section 2.8 in a natural way to prefix-classifier automata. Let $\mathcal{D}_W = (Q, C, q_{\text{init}}, \delta, F)$ be a prefix-classifier automaton. First, for $q \in Q$, we write $q^{-1}W$ for the set $w^{-1}W$, where w is any finite word such that $\delta^*(q_{\text{init}}, w) = q$. Secondly, there is a preorder $\preceq$ on the states of $\mathcal{D}_W$ based on the winning continuations of each state.

7.3 Safety objectives and monotony

We start our tour of regular objectives with general safety objectives, as they turn out to be simpler to understand than general reachability objectives as far as memory requirements are concerned. We reuse and revisit the $\mathcal{M}$-strong-monotony notion introduced in Subsection 4.3.2. To keep chapters as independent as possible, we briefly recall the interest of this notion here, and we recall one way to formulate its definition that will be relevant to the point of view assumed in this chapter.

Let us fix an objective $W \subseteq C^\omega$. In order to suffice to play optimally for $\mathcal{P}_1$ for W, a memory structure $\mathcal{M}$ needs to be able to distinguish between histories that are not comparable for the prefix preorder $\preceq$ of W. Indeed, if two finite words $w_1, w_2 \in C^*$ are not comparable, we can construct an arena in which the opponent chooses between playing w_1 and playing w_2, and then the correct choice has to be made between a continuation only winning after w_1, and a continuation only winning after w_2. This motivates the following property, which we showed to be equivalent to *$\mathcal{M}$-strong-monotony* in Chapter 4. We will not use the original definition (Definition 4.3.7) in this chapter; we will consider this reformulation as the definition of $\mathcal{M}$-strong-monotony.

Lemma 4.3.8 *Let $W \subseteq C^\omega$ be an objective and $\mathcal{M} = (M, m_{\text{init}}, \alpha_{\text{upd}})$ be a memory structure. Objective W is $\mathcal{M}$-strongly-monotone if and only if for all $m \in M$, for all $w, w' \in L_{m_{\text{init}}, m}$, w and w' are comparable for the prefix preorder $\leq$ of W.*

In short, W is $\mathcal{M}$-strongly-monotone if the memory structure $\mathcal{M}$ is able to keep track, not necessarily of the precise equivalence class of prefixes of the current finite word, but of a *chain* of prefixes in which that word lies. To illustrate this property, we give two particular cases.

▶ An objective with a *total* prefix preorder is $\mathcal{M}$-strongly-monotone for any structure $\mathcal{M}$: all pairs of finite words are comparable, and there is no need to have multiple states to distinguish them.

▶ A regular reachability or safety objective derived from a DFA $\mathcal{D}$ is $\mathcal{D}$-strongly-monotone (where we see $\mathcal{D}$ as a memory structure, ignoring its final states): if we keep track of the state of the DFA, we always know the precise equivalence class of the current history. Using $\mathcal{D}$ as a memory structure may have more states than necessary (as we just need to know a *chain* and not the precise equivalence class), but it is always sufficient.

We recall that W is $\mathcal{M}$-strongly-monotone if and only if $\overline{W}$ is $\mathcal{M}$-strongly-monotone, as being comparable for $\leq_W$ is equivalent to being comparable for $\leq_{\overline{W}} = \geq_W$ (Lemma 4.3.9).

The discussion above implies that for any objective W, $\mathcal{M}$-strong-monotony is necessary for a memory structure $\mathcal{M}$ to suffice to play optimally. We prove it formally, and we distinguish the regular from the general case: intuitively, regularity allows distinguishing distinct objectives with ultimately periodic words (Lemma 2.7.12), which can be encoded into finite arenas.

Proposition 7.3.1 (Necessity of $\mathcal{M}$-strong-monotony) *Let $W \subseteq C^\omega$ be an objective and $\mathcal{M}$ be a memory structure.*

1. *If W is ω-regular and $\mathcal{M}$ suffices to play optimally for $\mathcal{P}_1$ in all finite one-player arenas, then W is $\mathcal{M}$-strongly-monotone.*
2. *In general, if $\mathcal{M}$ suffices to play optimally for $\mathcal{P}_1$ in all finitely branching one-player arenas, then W is $\mathcal{M}$-strongly-monotone.*

Proof. Let $\mathcal{M} = (M, m_{\text{init}}, \alpha_{\text{upd}})$. We prove both items simultaneously, simply adding an observation in the regular case. We assume by contrapositive that W is not $\mathcal{M}$-strongly-monotone, i.e., there exist $w_1, w_2 \in C^*$ such that $\alpha_{\text{upd}}^*(m_{\text{init}}, w_1) = \alpha_{\text{upd}}^*(m_{\text{init}}, w_2)$, but w_1 and w_2 are not comparable for the prefix preorder $\leq$ of W. This means that there exist $w_1', w_2' \in C^\omega$ such that $w_1' \in w_1^{-1}W \setminus w_2^{-1}W$ and $w_2' \in w_2^{-1}W \setminus w_1^{-1}W$, i.e., such that $w_1 w_1' \in W$, $w_2 w_1' \notin W$, $w_2 w_2' \in W$, and $w_1 w_2' \notin W$. In case W is regular, then $w_1^{-1}W$ and $w_2^{-1}W$ are ω-regular, so we may assume additionally that there exist
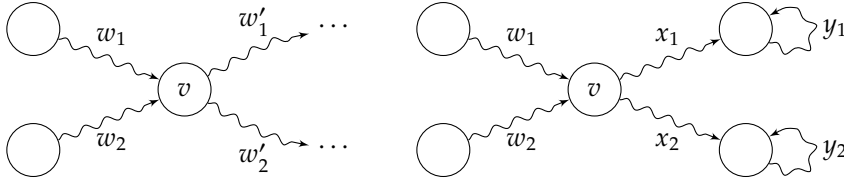
$x_1, x_2 \in C^*$ and $y_1, y_2 \in C^+$ such that $w'_1 = x_1(y_1)^\omega$ and $w'_2 = x_2(y_2)^\omega$ are ultimately periodic words (Lemma 2.7.12).

We build a one-player arena $\mathcal{A}$ in which $\mathcal{M}$ does not suffice to play optimally for $\mathcal{P}_1$: arena $\mathcal{A}$ is finitely branching in general, and can even be made finite when W is regular. In $\mathcal{A}$, there is a single vertex v in which a choice between two edges has to be made. This vertex v can be reached after seeing either w_1 or w_2, and the choice has to be made between continuing with the word w'_1 or with the word w'_2. We depict this arena in Figure 7.2.

An optimal strategy of $\mathcal{P}_1$ wins after seeing w_1 by continuing with w'_1, and after seeing w_2 by continuing with w'_2. However, a strategy based on $\mathcal{M}$ will make the same choice after seeing both w_1 and w_2 since $\alpha^*_{\mathsf{upd}}(m_{\mathsf{init}}, w_1) = \alpha^*_{\mathsf{upd}}(m_{\mathsf{init}}, w_2)$, and can therefore not be optimal. $\qquad\square$

In the case of general reachability or safety objectives, it is valuable to reformulate the notion of $\mathcal{M}$-strongly-monotone objectives using chains. Given a general reachability or safety objective W, its (possibly infinite) prefix-classifier automaton $\mathcal{D}_W = (Q, C, q_{\mathsf{init}}, \delta, F)$, and a memory structure $\mathcal{M} = (M, m_{\mathsf{init}}, \alpha_{\mathsf{upd}})$, we can associate with each state $m \in M$ the set $\Gamma_m^W \subseteq Q$ of states of $\mathcal{D}_W$ that can be reached "simultaneously to m". Formally, for $m \in M$,

$$\Gamma_m^W = \{\delta^*(q_{\mathsf{init}}, w) \in Q \mid w \in C^*, \alpha^*_{\mathsf{upd}}(m_{\mathsf{init}}, w) = m\}.$$

We drop the superscript W if there is no ambiguity. The following property then follows from Lemma 4.3.8, providing yet another reformulation of $\mathcal{M}$-strong-monotony.

> **Lemma 7.3.2** *Let $W \subseteq C^\omega$ be a general reachability or safety objective and $\mathcal{M} = (M, m_{\mathsf{init}}, \alpha_{\mathsf{upd}})$ be a memory structure. Objective W is $\mathcal{M}$-strongly-monotone if and only if for all $m \in M$, the set Γ_m is a chain for the prefix preorder $\leq$ of W.*

The formulation of $\mathcal{M}$-strong-monotony in Lemma 4.3.8 required that any two finite words reaching the same state of $\mathcal{M}$ must be comparable; here, we focus instead on the prefix-classifier automaton of W and require that states of the automaton that can be reached along with the same state of $\mathcal{M}$ are comparable.

Our first characterization states that for general safety objectives, $\mathcal{M}$-strong-monotony also implies that $\mathcal{M}$ suffices to play optimally. We state two variants of the results: in the first one, we assume that the preorder $\preceq$ induced by the objective is well-founded (which includes the regular case), and the result holds for all arenas; in the second one, we make no such assumption, but the result holds only for finitely branching arenas. We will discuss why we do not have the result with none of these hypotheses in Remark 7.3.6.

> **Theorem 7.3.3** (Characterization for safety) *Let $W \subseteq C^\omega$ be a general safety objective and $\mathcal{M}$ be a memory structure.*
>
> 1. *If the prefix preorder $\preceq$ of W is well-founded (in particular, if W is regular), then $\mathcal{M}$ suffices to play optimally for $\mathcal{P}_1$ if and only if W is $\mathcal{M}$-strongly-monotone.*
> 2. *In the general case, $\mathcal{M}$ suffices to play optimally for $\mathcal{P}_1$ in all finitely branching arenas if and only if W is $\mathcal{M}$-strongly-monotone.*

We prove this theorem in Subsection 7.6.1.

In particular, we find that $\mathcal{M}_{\mathsf{triv}}$ suffices (i.e., memoryless strategies suffice) for general safety objectives if and only if $\preceq$ is a *total* preorder, which was already a corollary of [CFH14] (chaotic and chromatic memoryless strategies coincide).

[CFH14]: Colcombet et al. (2014), *Playing Safe*

This characterization has an interest in itself (it covers a large class of relatively simple objectives), but as can be hinted by the one-to-two-player lift from Chapter 4 (Theorem 4.4.4), it will also be a useful technical step to understand the memory requirements of their complement, namely, the general reachability objectives (Section 7.4).

> **Remark 7.3.4** Memory structures are defined as finite throughout the thesis, but the previous Theorem 7.3.3 happens to work even with memory structures $\mathcal{M}$ with an *infinite* state space — we discuss this in its complete proof.

A corollary of this characterization, by comparing to the characterization for chaotic memory in [CFH14], is that chromatic and chaotic memory requirements differ already for regular safety objectives. We provide an instructive example below. This provides a new simple example answering the question of Kopczyński's [Kop08] mentioned in Subsection 2.6.2, which Casares [Cas22] first solved with a Muller objective.

[Kop08]: Kopczyński (2008), *Half-positional Determinacy of Infinite Games*

[Cas22]: Casares (2022), *On the Minimisation of Transition-Based Rabin Automata and the Chromatic Memory Requirements of Muller Conditions*
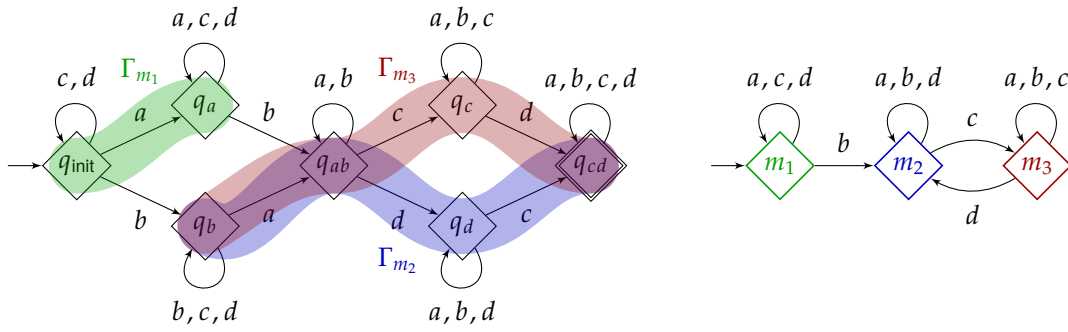
> **Example 7.3.5** Let $C = \{a, b, c, d\}$. We consider the regular language of finite words that first have to see both a and b (in any order), and only then see both c and d (in any order). A minimal DFA $\mathcal{D}$ recognizing this language is depicted in Figure 7.3 (left). We write W for the induced regular safety objective: $W = \mathsf{GenSafe}(\mathcal{L}(\mathcal{D}))$.

Figure 7.3: Example 7.3.5: DFA $\mathcal{D}$ (left) and a minimal memory structure $\mathcal{M}$ (right) such that GenReach($\mathcal{L}(\mathcal{D})$) and GenSafe($\mathcal{L}(\mathcal{D})$) are $\mathcal{M}$-strongly-monotone.

The main claim is that the chaotic memory requirements for W are two states, which is easily obtained from the existing characterization [CFH14] (this is the cardinality of a maximal antichain for $\preceq$), while the chromatic requirements for W are three states. We depict a memory structure $\mathcal{M}$ with three states which makes W $\mathcal{M}$-strongly-monotone in Figure 7.3 (right). To check that W is indeed $\mathcal{M}$-strongly-monotone, we have to check that there is no pair of words $w_1, w_2 \in C^*$ such that w_1 and w_2 reach the same state of $\mathcal{M}$, but reach non-comparable states in $\mathcal{D}$. The only two pairs of non-comparable states in $\mathcal{D}$ are q_a and q_b, and q_c and q_d (besides these, states are ordered for $\preceq$ from right to left). We can check that for this choice of $\mathcal{M}$, $\Gamma_{m_1} = \{q_{\text{init}}, q_a\}$, $\Gamma_{m_2} = \{q_b, q_{ab}, q_d, q_{cd}\}$, $\Gamma_{m_3} = \{q_b, q_{ab}, q_c, q_{cd}\}$. As these are all chains for $\preceq$, we have that W is $\mathcal{M}$-strongly-monotone.

It is not possible to find a chromatic memory structure $\mathcal{M}$ with *two* states that makes W $\mathcal{M}$-strongly-monotone (this can be checked by exhaustion, by trying to assign transitions to two states while distinguishing non-comparable states, and observing that all cases fail).

[CFH14]: Colcombet et al. (2014), *Playing Safe*

We finally discuss why, with neither the well-foundedness hypothesis nor the finitely branching hypothesis from Theorem 7.3.3, we cannot expect such a characterization.

Remark 7.3.6 If the prefix preorder of an objective W is not well-founded, then there is an infinite decreasing sequence of finite words $w_1 > w_2 > \ldots$ in C^*. This means that for all $i \geq 1$, there is $w_i' \in C^\omega$ such that $w_i w_i' \in W$, but for $j > i$, $w_j w_i' \notin W$. We can then build the infinitely branching arena depicted in Figure 7.4 in which $\mathcal{P}_2$ first chooses a word w_j, and $\mathcal{P}_1$ can win by playing a word w_i' with $i \geq j$. This requires infinite memory, even if W is $\mathcal{M}_{\text{triv}}$-strongly-monotone.

We may wonder whether such examples exist for general safety objectives; this is indeed the case. For instance, for $C = \{-1, 1, a\}$, the objective "avoid seeing a while the sum of colors from $\{-1, 1\}$ is negative" is a general safety objective with an infinite decreasing sequence of finite words (such as $((-1)^n)_{n \in \mathbb{N}}$).
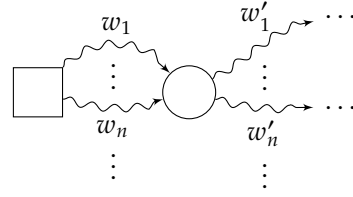
This construction is similar to the proof of Lemma 5.5.2.

7.4 Reachability objectives and progress

7.4.1 Capturing progress

To play optimally for general and regular reachability objectives with a memory structure $\mathcal{M}$, $\mathcal{M}$-strong-monotony is still necessary (Proposition 7.3.1) but this time not sufficient: the following example shows that the memory structure must keep track of *progress*.

> **Example 7.4.1** Let $C = \{a, b\}$. We consider the regular language $b^*a^+bC^*$ of words that have to see at least one a, followed by at least one b. This language is recognized by the DFA $\mathcal{D}$ in Figure 7.5 (left). We write W for the induced regular reachability objective: $W = \mathsf{GenReach}(\mathcal{L}(\mathcal{D}))$. In the arena in Figure 7.5 (center), $\mathcal{P}_1$ may win by starting a play with ab, but not without memory. The intuition is that playing a first makes some progress (it reaches an automaton state with more winning continuations), but is not sufficient to win, even if repeated. Therefore, in our memory structures, if a word makes some progress but without guaranteeing the win when repeated, we want the memory state to change upon reading that word. The memory structure in Figure 7.5 (right) is sufficient for W; in particular, seeing the first a, which makes progress from q_{init} to q_a, changes the memory state.

We formalize this intuition in the following definition.

> **Definition 7.4.2** ($\mathcal{M}$-progress-consistency) *Let $W \subseteq C^\omega$ be an objective and $\mathcal{M} = (M, m_{\mathsf{init}}, \alpha_{\mathsf{upd}})$ be a memory structure. We say that W is $\mathcal{M}$-progress-consistent if for all $m \in M$, for all $w_1 \in L_{m_{\mathsf{init}}, m}$, for all $w_2 \in L_{m,m}$, if $w_1 < w_1 w_2$, then $w_1 (w_2)^\omega \in W$.*

Intuitively, this says that if it is possible to come back to the same memory state while reading a "word that makes progress" (i.e., that improves our situation by putting us in a position with more winning continuations),
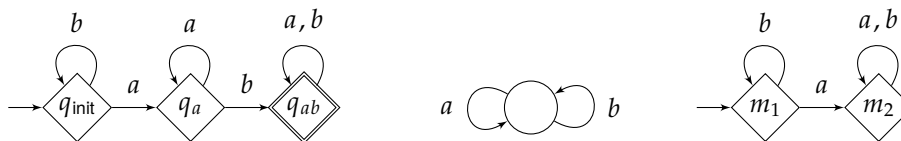
Figure 7.5: Example 7.4.1: DFA $\mathcal{D}$ (left), an arena requiring memory for $\mathsf{GenReach}(\mathcal{L}(\mathcal{D}))$ (center), and a minimal sufficient memory structure (right).

then repeating this word infinitely often from that point onward must be winning.

As for all the previous "$\mathcal{M}$-something" properties, we have that $\mathcal{M}$-progress-consistency is stable by memory product.

> **Lemma 7.4.3** *Let $\mathcal{M}$ be a memory structure and W be an $\mathcal{M}$-progress-consistent objective. Then, for all memory structures $\mathcal{M}'$, W is $(\mathcal{M} \otimes \mathcal{M}')$-progress-consistent.*

Proof. Let $\mathcal{M}' = (M', m'_{\text{init}}, \alpha'_{\text{upd}})$ be a memory structure. The state space of $\mathcal{M} \otimes \mathcal{M}'$ is $M \times M'$; let $(m, m') \in M \times M'$. The sets involved in the definition of $(\mathcal{M} \otimes \mathcal{M}')$-progress-consistency are "smaller" than the ones involved in the definition of $\mathcal{M}$-progress-consistency: by definition, $L_{(m_{\text{init}}, m'_{\text{init}}),(m,m')} \subseteq L_{m_{\text{init}},m}$ and $L_{(m,m'),(m,m')} \subseteq L_{m,m}$. Hence, as the definition of $(\mathcal{M} \otimes \mathcal{M}')$-progress-consistency quantifies universally over these sets, W is also $(\mathcal{M} \otimes \mathcal{M}')$-progress-consistent. $\qquad\square$

The discussion above (illustrated by Example 7.4.1) hints that for any objective W, $\mathcal{M}$-progress-consistency is necessary for a memory structure $\mathcal{M}$ to be sufficient to play optimally. As for $\mathcal{M}$-strong-monotony, we distinguish the regular case from the general case.

> **Proposition 7.4.4** (Necessity of $\mathcal{M}$-progress-consistency) *Let $W \subseteq C^\omega$ be an objective and $\mathcal{M}$ be a memory structure.*
>
> 1. *If W is ω-regular and $\mathcal{M}$ suffices to play optimally for $\mathcal{P}_1$ in all finite one-player arenas, then W is $\mathcal{M}$-progress-consistent.*
> 2. *In the general case, if $\mathcal{M}$ suffices to play optimally for $\mathcal{P}_1$ in all finitely branching one-player arenas, then W is $\mathcal{M}$-progress-consistent.*

Proof. Let $\mathcal{M} = (M, m_{\text{init}}, \alpha_{\text{upd}})$. We prove both items simultaneously. We assume by contrapositive that W is not $\mathcal{M}$-progress-consistent, i.e., there exist $m \in M$, $w_1 \in L_{m_{\text{init}},m}$, and $w_2 \in L_{m,m}$ such that $w_1 \prec w_1 w_2$ but $w_1(w_2)^\omega \notin W$. As $w_1 \prec w_1 w_2$, there is $w' \in C^\omega$ such that $w_1 w' \notin W$ and $w_1 w_2 w' \in W$. In case W is ω-regular, then $w_1^{-1} W$ and $(w_1 w_2)^{-1} W$ are ω-regular, so we may assume additionally that they are distinguished by an ultimately periodic word (Lemma 2.7.12): there exist $x \in C^*$ and $y \in C^+$ such that $w' = xy^\omega$.

We build a one-player arena $\mathcal{A}$ in which $\mathcal{M}$ does not suffice to play optimally for $\mathcal{P}_1$: arena $\mathcal{A}$ is finitely branching in general, and can even be made finite when W is ω-regular. In $\mathcal{A}$, there is a single vertex v in which a choice between two edges has to be made. This vertex can be reached after seeing w_1, and the choice has to be made between looping on v with word w_2, or continuing with word w'. We depict this arena in Figure 7.6.
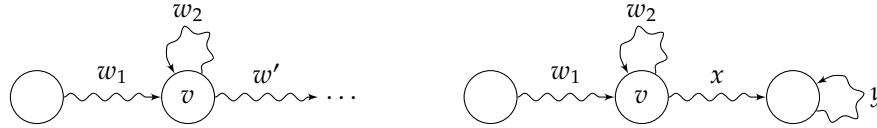
Figure 7.6: Arena in which $\mathcal{P}_1$ cannot play optimally with a strategy based on $\mathcal{M}$ obtained from the proof of Proposition 7.4.4. The arena on the left is used in the general case, and the one on the right is used in the regular case.

An optimal strategy of $\mathcal{P}_1$ wins after seeing w_1 by continuing with $w_2 w'$, which produces the winning word $w_1 w_2 w'$. However, a strategy based on $\mathcal{M}$ must always make the same choice in v after seeing w_1 since $\alpha^*_{\mathsf{upd}}(m_{\mathsf{init}}, w_1) = \alpha^*_{\mathsf{upd}}(m_{\mathsf{init}}, w_1(w_2)^n) = m$ for all $n \geq 0$. Hence, a strategy based on $\mathcal{M}$ can only produce losing words $w_1 w'$ and $w_1(w_2)^\omega$. □

The following example should help the reader form the right intuition about $\mathcal{M}$-progress-consistency.

Example 7.4.5 Let $C = \{a, b\}$. We consider the regular language of words containing *ababa* as a subword, recognized by the DFA $\mathcal{D}$ in Figure 7.7 (left). We consider the memory structure $\mathcal{M}$ remembering whether a or b was last seen, depicted in Figure 7.7 (right). The regular reachability objective $W = \mathsf{GenReach}(\mathcal{L}(\mathcal{D}))$ is $\mathcal{M}$-progress-consistent. Indeed, let us first consider $m = m_b$ in the definition of $\mathcal{M}$-progress-consistency. A finite word w_1 reaching m_b in $\mathcal{M}$ necessarily reaches q_{init}, q_{ab}, or q_{abab} in Q (excluding the final state from the reasoning, as no progress is possible from it). After w_1, words w_2 that both (*i*) make progress ($w_1 \prec w_1 w_2$) and (*ii*) are a cycle on m_b necessarily see both a and b. Therefore, $w_1(w_2)^\omega$ is always a winning word. The same reasoning holds for $m = m_a$. Notice that the memory states from the memory structure do not carry enough information to ascertain when a word of the language has been seen (i.e., when the game is won).

The upcoming Theorem 7.4.8 implies that $\mathcal{M}$ suffices to play optimally for $\mathcal{P}_1$.

This need to capture *progress* was not necessary to understand the memory requirements of safety objectives, which may be explained by the following reasoning.

Remark 7.4.6 Unlike general reachability objectives, all general *safety* objectives are $\mathcal{M}_{\mathsf{triv}}$-progress-consistent. Here is a proof of this statement. Let $W \subseteq C^\omega$ be a general safety objective. Let $w_1, w_2 \in L^{\mathcal{M}_{\mathsf{triv}}}_{m_{\mathsf{init}}, m_{\mathsf{init}}} = C^*$ be
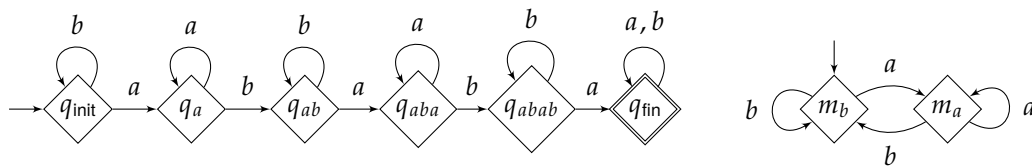

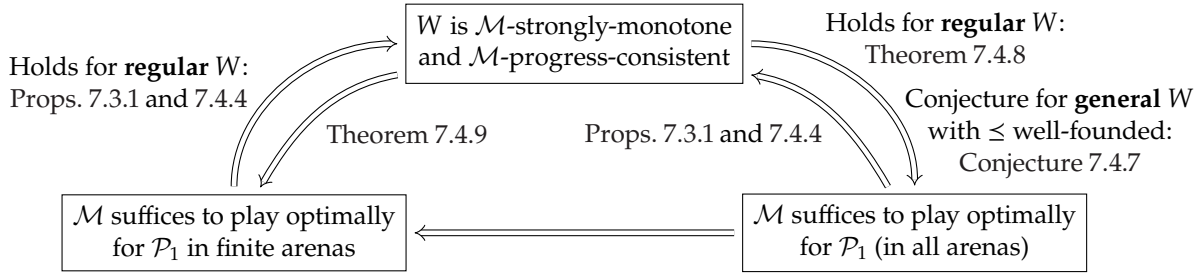**Figure 7.7:** Example 7.4.5: DFA $\mathcal{D}$ (left) and memory structure $\mathcal{M}$ (right).

Figure 7.8: Summary of the implications that hold and that we conjecture about the memory requirements of general and regular reachability objectives. Everything is equivalent for regular reachability objectives, but only some implications hold for general reachability objectives. Objective W is a general reachability objective and $\mathcal{M}$ is a memory structure.

such that $w_1 \prec w_1 w_2$. This implies that $w_1 w_2$, and therefore w_1, have a non-empty set of winning continuations. Assume by contradiction that $w_1 (w_2)^\omega \notin W$. As W is a general safety objective, there is a smallest $n \geq 1$ such that $w_1 (w_2)^n$ has no winning continuation. Hence, $w_1 (w_2)^{n-1}$ still has some winning continuations, so $w_1 (w_2)^n \prec w_1 (w_2)^{n-1}$. This is a contradiction, as $w_1 \prec w_1 w_2$ implies that $w_1 (w_2)^{n-1} \preceq w_1 w_2 (w_2)^{n-1} = w_1 (w_2)^n$ by Lemma 2.8.5. This property is, at least intuitively, a reason hinting that the memory requirements of safety objectives are lower and easier to study than those for their complement reachability objective.

7.4.2 Understanding memory requirements

We have now discussed two necessary properties for a memory structure $\mathcal{M}$ to be sufficient to play optimally for any objective. For general reachability objectives, it appears that the conjunction of these two properties is also sufficient in many situations.

We have to discuss here some slight limit to our results, which leads to characterizations not as complete as the one for general safety conditions (Theorem 7.3.3). A summary of the known and unknown implications about the memory requirements of general reachability objectives is provided in Figure 7.8.

Our initial intuition — which is not known to be true — is that $\mathcal{M}$-strong-monotony and $\mathcal{M}$-progress-consistency are equivalent to the sufficiency of $\mathcal{M}$ for general reachability objectives.

Conjecture 7.4.7 *Let $W \subseteq C^\omega$ be a general reachability objective and $\mathcal{M}$ be a memory structure.*

1. *If the prefix preorder $\leq$ of W is well-founded (in particular, if W is regular), then $\mathcal{M}$ suffices to play optimally for $\mathcal{P}_1$ if and only if W is $\mathcal{M}$-strongly-monotone and $\mathcal{M}$-progress-consistent.*
2. *In the general case, $\mathcal{M}$ suffices to play optimally for $\mathcal{P}_1$ in all finitely branching arenas if and only if W is $\mathcal{M}$-strongly-monotone and $\mathcal{M}$-progress-consistent.*

The need to restrict to a well-founded $\leq$ or to arenas with finite branching was argued in Remark 7.3.6.

Instead, we prove two incomparable results getting very close to this conjecture. First, we actually have the desired equivalence for regular reachability objectives.

> **Theorem 7.4.8** (Characterization for regular reachability) *Let $W \subseteq C^\omega$ be a* regular *reachability objective and $\mathcal{M}$ be a memory structure. Memory $\mathcal{M}$ suffices to play optimally for $\mathcal{P}_1$ if and only if W is $\mathcal{M}$-strongly-monotone and $\mathcal{M}$-progress-consistent.*

We defer the proof of this result to Subsection 7.6.2. We use here an ad hoc proof very similar to — but also more involved than — the proof of Theorem 7.3.3 for general safety conditions. As far as the computational complexity (Section 7.5) and the implementation (Section 7.7) are concerned, the above result suffices for our needs, as we only consider regular objectives.

Secondly, the two conditions suffice to play optimally in *finite* arenas, even for general reachability objectives.

> **Theorem 7.4.9** (Sufficiency in finite arenas) *Let $W \subseteq C^\omega$ be a general reachability objective and $\mathcal{M}$ be a memory structure. If W is $\mathcal{M}$-strongly-monotone and $\mathcal{M}$-progress-consistent, then $\mathcal{M}$ suffices to play optimally for $\mathcal{P}_1$ in finite arenas.*

The proof of this result is a relatively straightforward application of the arena-independent one-to-two-player lift from Chapter 4 (Theorem 4.4.4), which we give in the next subsection.

> **Remark 7.4.10** We will discuss in Subsection 7.6.2, Example 7.6.1 why our proof techniques do not apply to *general* reachability objectives played on *infinite* arenas, i.e., the setting of Conjecture 7.4.7 (even with well-founded prefix preorder $\leq$ and finite branching of the arenas).

Observe that, due to Remark 7.4.6 showing that all general safety objectives are $\mathcal{M}_{\text{triv}}$-progress-consistent, Conjecture 7.4.7 actually works if we replace "general reachability condition" by "general safety condition"; with general safety conditions, we simply always have $\mathcal{M}$-progress-consistency for free for all memory structures $\mathcal{M}$.

It is unclear how to represent classes of general reachability and safety objectives in a finite way, so we focus our implementation on the regular case.

7.4.3 Proof via one-to-two-player lift

To use the one-to-two-player lift (Theorem 4.4.4) with general reachability objectives, we need to understand both players' memory requirements in finite one-player arenas. We have already done so for the complement general safety objectives in Theorem 7.3.3, which solves the problem for $\mathcal{P}_2$. We still need to understand the memory requirements of $\mathcal{P}_1$ in its one-player arenas, which, as was shown in the "one-player characterization"

(Chapter 4, Theorem 4.4.3), boils down to the $\mathcal{M}$-*monotony* and $\mathcal{M}$-*selectivity* properties.

We already assume $\mathcal{M}$-strong-monotony in our hypotheses, which implies $\mathcal{M}$-monotony. The crux is to notice that $\mathcal{M}$-progress-consistency implies $\mathcal{M}$-selectivity under $\mathcal{M}$-strong-monotony for general reachability objectives (but not for all objectives, even the ω-regular ones).

> **Lemma 7.4.11** *Let $W \subseteq C^\omega$ be a general reachability objective, and $\mathcal{M}$ be a memory structure such that W is $\mathcal{M}$-strongly-monotone. If W is $\mathcal{M}$-progress-consistent, then W is $\mathcal{M}$-selective.*

Proof. We write $\mathcal{M} = (M, m_{\text{init}}, \alpha_{\text{upd}})$. Let $\mathcal{D}_W$ be the (possibly infinite) prefix-classifier automaton of W. We assume that W is $\mathcal{M}$-progress-consistent and show that it is $\mathcal{M}$-selective. Let $m \in M$, $x \in L_{m_{\text{init}}, m}$, and $K_1, K_2, K_3 \subseteq C^*$ be regular languages such that $K_1, K_2 \subseteq L_{m,m}$. Let $w \in x[(K_1 \cup K_2)^* K_3]$ be a winning word; we show that there is also a winning word in $x[K_1^*] \cup x[K_2^*] \cup x[K_3]$. We distinguish a few cases.

- ▶ If a prefix of x is in $\mathcal{L}(\mathcal{D})$, this is clear.
- ▶ As $x \in L_{m_{\text{init}}, m}$ and $K_1, K_2 \subseteq L_{m,m}$, we have that for all $y \in (K_1 \cup K_2)$, $\alpha_{\text{upd}}^*(m_{\text{init}}, x) = \alpha_{\text{upd}}^*(m_{\text{init}}, xy) = m$. As W is $\mathcal{M}$-strongly-monotone, for all $y \in (K_1 \cup K_2)$, x is comparable to xy for $\preceq$. If there is $y \in (K_1 \cup K_2)$ such that $x \prec xy$, then xy^ω is winning by $\mathcal{M}$-progress-consistency. Moreover, $xy^\omega \in x[K_1^*] \cup x[K_2^*]$, which ends this case. Otherwise,

$$\forall y \in (K_1 \cup K_2), xy \preceq x. \tag{7.1}$$

We work under this assumption for the remaining cases. We have that $x[(K_1 \cup K_2)^* K_3] = x[(K_1 \cup K_2)^*] \cup x(K_1 \cup K_2)^*[K_3]$; we will consider the two possibilities separately for w.

- • We assume that $w \in x[(K_1 \cup K_2)^*]$. As $w \in W$ and $W = \mathsf{GenReach}(\mathcal{L}(\mathcal{D}))$, there is a prefix $xy \in C^*$ of w that is in $\mathcal{L}(\mathcal{D})$. We have by definition that $y \in \mathsf{Prefs}((K_1 \cup K_2)^*)$, i.e., $y = y_1 \ldots y_n y_*$, where $y_i \in (K_1 \cup K_2)$ and $y_* \in \mathsf{Prefs}(K_1 \cup K_2)$. By induction and Equation (7.1), we can show that $xy_1 \ldots y_n \preceq x$. Therefore, word xy_* is also in $\mathcal{L}(\mathcal{D})$, and it is the prefix of an infinite word in $x[K_1^*] \cup x[K_2^*]$.
- • If $w \in x(K_1 \cup K_2)^*[K_3]$, then $w = xy_1 \ldots y_n z$, with $y_i \in (K_1 \cup K_2)$ and $z \in [K_3]$. As before, $xy_1 \ldots y_n \preceq x$, so $xz \in x[K_3]$ is also winning. □

For completeness, we also show that the converse implications ($\mathcal{M}$-selectivity implies $\mathcal{M}$-progress-consistency) holds for ω-regular objectives (Lemma 7.4.12). Along with Lemma 7.4.11, this shows that under $\mathcal{M}$-strong-monotony, $\mathcal{M}$-progress-consistency is equivalent to $\mathcal{M}$-selectivity for regular reachability objectives.

> **Lemma 7.4.12** *Let $W \subseteq C^\omega$ be an ω-regular objective, and $\mathcal{M}$ be a memory structure. If W is $\mathcal{M}$-selective, then W is $\mathcal{M}$-progress-consistent.*

Proof. Let $\mathcal{M} = (M, m_{\text{init}}, \alpha_{\text{upd}})$. We assume by contrapositive that W is not $\mathcal{M}$-progress-consistent, i.e., there exist $m \in M$, $w_1 \in L_{m_{\text{init}}, m}$, and $w_2 \in L_{m,m}$ such that $w_1 \prec w_1 w_2$ and $w_1 (w_2)^\omega \notin W$. Let $w_3 \in C^\omega$ be a word such that $w_1 w_3 \notin W$ and $w_1 w_2 w_3 \in W$, which exists as $w_1 \prec w_1 w_2$. Without loss of generality, we can assume that there exist $x, y \in C^*$ such that $w_3 = x y^\omega$, as two distinct ω-regular objectives can be distinguished by an ultimately periodic word (Lemma 2.7.12). Let $w = w_1 \in L_{m_{\text{init}}, m}$, $K_1 = \{w_2\} \subseteq L_{m,m}$, $K_2 = \emptyset \subseteq L_{m,m}$, and $K_3 = x y^*$. Sets K_1, K_2, and K_3 are regular. We have that $[w(K_1 \cup K_2)^* K_3]$ contains the winning word $w_1 w_2 w_3$. However, we have $[w K_1^*] = \{w_1 (w_2)^\omega\}$, $[w K_2^*] = \emptyset$, and $[w K_3] = \{w_1 w_3\}$: the set $[w K_1^*] \cup [w K_2^*] \cup [w K_3]$ contains only losing words. Hence, we do not have $\mathcal{M}$-selectivity. $\qquad\square$

We can now easily prove Theorem 7.4.9 about the sufficiency of $\mathcal{M}$-strong-monotony and $\mathcal{M}$-progress-consistency to play optimally in finite arenas, using the one-to-two-player lift from Chapter 4.

Proof of Theorem 7.4.9. We assume that W is $\mathcal{M}$-strongly-monotone and $\mathcal{M}$-progress-consistent. As W is $\mathcal{M}$-strongly-monotone, we also have that $\overline{W}$ is $\mathcal{M}$-strongly-monotone (Lemma 4.3.9). As $\overline{W}$ is a general safety objective, we have that $\mathcal{M}$ suffices for $\mathcal{P}_2$ by Theorem 7.3.3 (in particular, it suffices in the *finite one-player* arenas of $\mathcal{P}_2$).

We show that $\mathcal{M}$ also suffices for $\mathcal{P}_1$ in its finite one-player arenas. We already have that W is $\mathcal{M}$-strongly-monotone; hence, it is $\mathcal{M}$-monotone. As W is a general reachability objective that is $\mathcal{M}$-strongly-monotone and $\mathcal{M}$-progress-consistent, by Lemma 7.4.11, W is also $\mathcal{M}$-selective. By Theorem 4.4.3, we deduce that $\mathcal{M}$ suffices to play optimally in the finite one-player arenas of $\mathcal{P}_1$.

We now know that $\mathcal{M}$ suffices in the finite one-player arenas of both players. By Theorem 4.4.4, we have in particular that $\mathcal{M} \otimes \mathcal{M}$, which is isomorphic to $\mathcal{M}$, suffices for $\mathcal{P}_1$ in all finite (two-player) arenas. $\qquad\square$

> **Remark 7.4.13** By restricting our focus to finite arenas, we could also use the machinery from Chapter 4 for the regular safety objectives, but it would not yield tight memory requirements. Let $W \subseteq C^\omega$ be a regular safety objective. It is relatively straightforward to show that $\mathcal{M}$-strong-monotony of W implies $\mathcal{M}$-selectivity of W. This shows that a memory structure $\mathcal{M}$ suffices for $\mathcal{P}_1$ in finite *one*-player arenas if and only if W is $\mathcal{M}$-strongly-monotone. If W is $\mathcal{M}$-strongly-monotone, we can then use the one-to-two-player lift to deduce an upper bound on the memory requirements in two-player games given by $\mathcal{M} \otimes \mathcal{M}'$, where $\mathcal{M}'$ is a memory structure for $\mathcal{P}_2$. Unfortunately, this memory structure

$\mathcal{M}'$ needs to be sufficient for a regular reachability objective, and is in general larger than $\mathcal{M}$ (as the characterizations suggest: $\overline{W}$ needs to additionally satisfy $\mathcal{M}'$-progress-consistency). Therefore, this proof technique would not yield *tight* bounds on the memory requirements of general safety objectives.

However, it works to deduce tight bounds for general reachability objectives in Theorem 7.4.9, as a sufficient memory structure $\mathcal{M}'$ in one-player arenas for such objectives happens to also suffice for the other player, and the product structure $\mathcal{M}' \otimes \mathcal{M}'$ is isomorphic to $\mathcal{M}'$.

For objectives beyond reachability and safety, $\mathcal{M}$-strong-monotony and $\mathcal{M}$-progress-consistency may not imply the sufficiency of $\mathcal{M}$ to play optimally for a player. For instance, observe that prefix-independence of an objective immediately implies both $\mathcal{M}_{\text{triv}}$-strong-monotony and $\mathcal{M}_{\text{triv}}$-progress-consistency for both players and, clearly, not all prefix-independent objectives are memoryless-determined. Perhaps one of the easiest examples, with $C = \{a, b\}$, is the objective

$$W = \text{Büchi}(a) \cap \text{Büchi}(b)$$
$$= \{w \in C^\omega \mid a \text{ and } b \text{ are both seen infinitely often}\},$$

This example and closely related ones involving Büchi automata will be discussed in more detail in Chapter 8.

which is ω-regular (it can be recognized by a *deterministic Büchi automaton* with two states), but is not a general reachability nor safety objective. Objective W is $\mathcal{M}_{\text{triv}}$-strongly-monotone and $\mathcal{M}_{\text{triv}}$-progress-consistent, but $\mathcal{M}_{\text{triv}}$ does not suffice to play optimally (Example 2.5.3).

7.4.4 Stronger lift for regular objectives

As a by-product of our results, we observe that for *regular* objectives, our characterizations (Theorems 7.3.3 and 7.4.8) deal with arbitrary arenas of any cardinality, but the properties used in the characterizations are already necessary in *finite one-player* arenas. This means that strategy-wise, to achieve a regular objective, all the complexity already appears in finite graphs with no opponent. For the specific class of regular objectives from this chapter, this strengthens the one-to-two-player lift from Chapter 4 by going beyond two of its limitations:

- ▶ the limitation to a result about two-player games played on *finite* arenas;
- ▶ the need to know a sufficient memory structure in the one-player arenas of *each* player — here, each player can use the same memory structure in its one-player and in the two-player arenas, with no need for the "$\mathcal{M}_1 \otimes \mathcal{M}_2$" product from Theorem 4.4.4.

Theorem 7.4.14 (Finite-to-infinite, one-to-two-player lift for regular objectives) *Let $W \subseteq C^\omega$ be a regular (reachability or safety) objective and $\mathcal{M}$*

> *be a memory structure. Memory $\mathcal{M}$ suffices to play optimally for $\mathcal{P}_1$ (in all arenas) if and only if $\mathcal{M}$ suffices to play optimally for $\mathcal{P}_1$ in its finite one-player arenas.*

Proof. The implication from left-to-right holds as this is the same property quantified over fewer arenas. We argue the other implication for each case.

For regular safety objectives W, we showed that if $\mathcal{M}$ suffices in finite one-player arenas, then W is $\mathcal{M}$-strongly-monotone (by Proposition 7.3.1 as W is regular), which implies that $\mathcal{M}$ suffices in all arenas (by Theorem 7.3.3 as W is a safety condition with a well-founded preorder).

For regular reachability objectives W, we showed that if $\mathcal{M}$ suffices in finite one-player arenas, then W is $\mathcal{M}$-strongly-monotone and $\mathcal{M}$-progress-consistent (by Propositions 7.3.1 and 7.4.4 as W is regular), which implies that $\mathcal{M}$ suffices in all arenas (by Theorem 7.4.8 as W is a regular reachability objective). □

7.5 The complexity of finding small memory structures

We discuss the computational complexity of finding small memory structures for regular objectives. We formalize the question as two decision problems: given a regular reachability or safety objective, how much memory is required to play optimally for $\mathcal{P}_1$?

Memory-Safe

Input: A DFA $\mathcal{D}$ inducing the regular safety objective $W = \mathsf{GenSafe}(\mathcal{L}(\mathcal{D}))$ and an integer $k \in \mathbb{N}$.

Question: Does there exist a memory structure $\mathcal{M}$ with at most k states which suffices to play optimally for $\mathcal{P}_1$?

Memory-Reach

Input: A DFA $\mathcal{D}$ inducing the regular reachability objective $W = \mathsf{GenReach}(\mathcal{L}(\mathcal{D}))$ and an integer $k \in \mathbb{N}$.

Question: Does there exist a memory structure $\mathcal{M}$ with at most k states which suffices to play optimally for $\mathcal{P}_1$?

It follows from our characterizations (Theorems 7.3.3 and 7.4.8) that the first problem is equivalent to asking whether there is a memory structure $\mathcal{M}$ with at most k states such that $\mathsf{GenSafe}(\mathcal{L}(\mathcal{D}))$ is $\mathcal{M}$-strongly-monotone, and the second problem whether there is a memory structure $\mathcal{M}$ with at most k states such that $\mathsf{GenReach}(\mathcal{L}(\mathcal{D}))$ is $\mathcal{M}$-strongly-monotone and $\mathcal{M}$-progress-consistent.

Theorem 7.5.1 (Complexity of Memory-Safe and Memory-Reach) *Both Memory-Safe and Memory-Reach are NP-complete.*

We defer the proof to Subsection 7.6.3. For NP-hardness, we construct a reduction from the Hamiltonian cycle problem which works for both Memory-Safe and Memory-Reach.

Our main insight is to reformulate the $\mathcal{M}$-strong-monotony notion (NP-membership of Memory-Safe follows from this reformulation). Let $W = \mathsf{GenSafe}(\mathcal{L}(\mathcal{D}))$ be a regular objective and $\mathcal{M} = (M, m_{\mathsf{init}}, \alpha_{\mathsf{upd}})$ be a memory structure. In Example 7.3.5, we have seen how to go from a memory structure $\mathcal{M}$ such that W is $\mathcal{M}$-strongly-monotone to a covering of the states of $\mathcal{D}$ by chains of states. We formulate exactly the requirements for such coverings in order to have a point of view equivalent to $\mathcal{M}$-strong-monotony. For $\Gamma \subseteq Q$ a set of automaton states and $c \in C$ a color, we define $\delta(\Gamma, c) = \{\delta(q, c) \mid q \in \Gamma\}$.

Definition 7.5.2 (Monotone decomposition) *Let $\mathcal{D} = (Q, C, q_{\mathsf{init}}, \delta, F)$ be a (possibly infinite) automaton. We say that the sets $\Gamma_1, \dots, \Gamma_k \subseteq Q$ form a monotone decomposition of $\mathcal{D}$ if*

  (a) $Q = \bigcup_{i=1}^k \Gamma_i$,
  (b) for all $c \in C$, for all $i \in \{1, \dots, k\}$, there is $j \in \{1, \dots, k\}$ such that $\delta(\Gamma_i, c) \subseteq \Gamma_j$, and
  (c) for all $i \in \{1, \dots, k\}$, Γ_i is a chain for $\leq$.

Note that the sets Γ_i do not have to be disjoint (and even *cannot* be disjoint to obtain minimal monotone decompositions, as was illustrated in Example 7.3.5). We invite the reader to verify the three requirements on the covering drawn in Figure 7.3 (left).

If we only consider requirements (a) and (b) of this definition, we recover the definition of an *admissible decomposition*, which can be used to quotient an automaton [GY65]. Here, we add the additional requirement (c) that each set of states is a chain for $\leq$. Note that there always exists an admissible decomposition with just one set (by taking $\Gamma_1 = Q$), but finding a small *monotone* decomposition may not be so easy. This point of view in terms of monotone decompositions turns out to be equivalent to our initial point of view in terms of $\mathcal{M}$-strong-monotony in the following sense.

[GY65]: Ginzburg et al. (1965), *Products of Automata and the Problem of Covering*

Lemma 7.5.3 *Let $\mathcal{D}$ be a (possibly infinite) automaton and $W \subseteq C^\omega$ be equal to $\mathsf{GenSafe}(\mathcal{L}(\mathcal{D}))$ or $\mathsf{GenReach}(\mathcal{L}(\mathcal{D}))$. Automaton $\mathcal{D}$ admits a monotone decomposition with $k \in \mathbb{N}$ sets if and only if W is $\mathcal{M}$-strongly-monotone for some memory structure $\mathcal{M}$ with k states.*

Proof. We write $\mathcal{D} = (Q, C, q_{\mathsf{init}}, \delta, F)$. Starting from a monotone decomposition $\Gamma_1, \dots, \Gamma_k$ of $\mathcal{D}$, we show how to build a memory structure

$\mathcal{M} = (M, m_{\mathsf{init}}, \alpha_{\mathsf{upd}})$ with k states such that W is $\mathcal{M}$-strongly-monotone. We take

- $M = \{\Gamma_1, \ldots, \Gamma_k\}$,
- m_{init} is any set Γ_i that contains q_{init} (which exists as $Q = \bigcup_{i=1}^k \Gamma_i$),
- for $\Gamma_i \in M$, $c \in C$, we define $\alpha_{\mathsf{upd}}(\Gamma_i, c) = \Gamma_j$ for some j such that $\delta(\Gamma_i, c) \subseteq \Gamma_j$ (there may be multiple such j's; any choice works).

We first show the following property about $\mathcal{M}$: for all $w \in C^*$, $\delta^*(q_{\mathsf{init}}, w) \in \alpha_{\mathsf{upd}}^*(m_{\mathsf{init}}, w)$. We proceed by induction on the length of w. If $w = \varepsilon$ is the empty word, then $\delta^*(q_{\mathsf{init}}, w) = q_{\mathsf{init}} \in m_{\mathsf{init}} = \alpha_{\mathsf{upd}}^*(m_{\mathsf{init}}, w)$ by definition of m_{init}. We now assume that $w = w'c$, with $c \in C$ and $\delta^*(q_{\mathsf{init}}, w') \in \alpha_{\mathsf{upd}}^*(m_{\mathsf{init}}, w')$. Let $\Gamma_i = \alpha_{\mathsf{upd}}^*(m_{\mathsf{init}}, w')$ and $\Gamma_j = \alpha_{\mathsf{upd}}^*(m_{\mathsf{init}}, w'c)$. Then, $\delta^*(q_{\mathsf{init}}, w'c) \in \delta(\Gamma_i, c)$. As $\delta(\Gamma_i, c) \subseteq \Gamma_j = \alpha_{\mathsf{upd}}^*(m_{\mathsf{init}}, w'c)$, we are done.

We now show that W is $\mathcal{M}$-strongly-monotone. Let $w_1, w_2 \in C^*$ be two finite words such that $\alpha_{\mathsf{upd}}^*(m_{\mathsf{init}}, w_1) = \alpha_{\mathsf{upd}}^*(m_{\mathsf{init}}, w_2)$. We set $\Gamma_i = \alpha_{\mathsf{upd}}^*(m_{\mathsf{init}}, w_1)$. We need to show that w_1 and w_2 are comparable for $\preceq$. Let $q_1 = \delta^*(q_{\mathsf{init}}, w_1)$ and $q_2 = \delta^*(q_{\mathsf{init}}, w_2)$. By the above property, we have that q_1 and q_2 are in Γ_i. As Γ_i is a chain, we have that q_1 and q_2 are comparable for $\preceq$. Hence, w_1 and w_2 are too, which shows the desired implication.

Reciprocally, let $\mathcal{M} = (M, m_{\mathsf{init}}, \alpha_{\mathsf{upd}})$ be a memory structure such that W is $\mathcal{M}$-strongly-monotone. We show that sets $(\Gamma_m)_{m \in M}$ form a monotone decomposition of $\mathcal{D}$.

(a) As $\mathcal{M}$ is a complete structure, every (reachable) state q of $\mathcal{D}$ has to be in a set Γ_m for some $m \in M$. Indeed, as there exists $w \in C^*$ such that $\delta^*(q_{\mathsf{init}}, w) = q$, we can simply take $m = \alpha_{\mathsf{upd}}^*(m_{\mathsf{init}}, w)$.

(b) Let $c \in C$ and $m \in M$. Let $m' = \alpha_{\mathsf{upd}}(m, c)$. We show that $\delta(\Gamma_m, c) \subseteq \Gamma_{m'}$. Let $q \in \Gamma_m$; we show that $\delta(q, c) \in \Gamma_{m'}$. As $q \in \Gamma_m$, there is $w \in C^*$ such that $\delta^*(q_{\mathsf{init}}, w) = q$ and $\alpha_{\mathsf{upd}}^*(m_{\mathsf{init}}, w) = m$. Then, $\delta^*(q_{\mathsf{init}}, wc) = \delta(q, c)$ and $\alpha_{\mathsf{upd}}^*(m_{\mathsf{init}}, wc) = m'$, so $\delta(q, c) \in \Gamma_{m'}$.

(c) For some $m \in M$, let $q_1, q_2 \in \Gamma_m$. We show that q_1 and q_2 are comparable for $\preceq$. There are words $w_1, w_2 \in C$ such that $\delta^*(q_{\mathsf{init}}, w_1) = q_1$, $\delta^*(q_{\mathsf{init}}, w_2) = q_2$, and $\alpha_{\mathsf{upd}}^*(m_{\mathsf{init}}, w_1) = \alpha_{\mathsf{upd}}^*(m_{\mathsf{init}}, w_2) = m$. As W is $\mathcal{M}$-strongly-monotone, w_1 and w_2 are comparable for $\preceq$, so that is also the case for q_1 and q_2. This shows that all sets Γ_m are chains. $\square$

> **Remark 7.5.4** For an automaton $\mathcal{D} = (Q, C, q_{\mathsf{init}}, \delta, F)$, there is always a monotone decomposition with $|Q|$ sets, which can be achieved by taking the family of sets $(\{q\})_{q \in Q}$ (it clearly satisfies the three requirements from Definition 7.5.2). In the equivalence from Lemma 7.5.3 above, this monotone decomposition corresponds to taking $\mathcal{D}$ (syntactically, after removing its set of final states from the tuple) as a memory structure. A slightly improved general upper bound for the size of monotone decompositions is given by $|Q| - 1$ sets. We assume w.l.o.g. that $\mathcal{D}$ has a single, absorbing final state q_{fin}. We can then consider the monotone decomposition $(\{q, q_{\mathsf{fin}}\})_{q \in Q \setminus \{q_{\mathsf{fin}}\}}$. In the equivalence from Lemma 7.5.3

The fact that $\mathcal{D}$ can be assumed to have a single absorbing final state was argued after Definition 7.2.1 on page 157.

above, this monotone decomposition corresponds to taking $\mathcal{D}$ without state q_{fin} as a memory structure — transitions to q_{fin} can be redirected to any state. This corresponds to the intuition that there is no point in keeping track of what happens once the game has been won (in the case of general reachability objectives) or lost (in the case of general safety objectives). This construction actually yields the optimal bound for conjunctions of simple reachability objectives [FH10].

It is instructive to reformulate the characterization of *chaotic* memory requirements from [CFH14]: the original phrasing was that the number of memory states necessary and sufficient to play optimally for the safety objective W is the cardinality of the largest antichain of its prefix preorder. Using our terminology and Dilworth's theorem [Dil50], it is equivalent to the smallest number of chains required to cover all states; that is, decompositions satisfying (a) and (c) in Definition 7.5.2, but not necessarily (b). Hence, it is smaller in general.

We have yet to discuss membership in NP of MEMORY-REACH, which is slightly more involved and is explained in Subsection 7.6.3, Lemma 7.6.3. We can reduce $\mathcal{M}$-progress-consistency to checking a polynomial number of emptiness queries of intersections of regular languages recognized by deterministic finite automata. For a given DFA $\mathcal{D} = (Q, C, q_{\text{init}}, \delta, F)$ and a witness memory structure $\mathcal{M}$, $\mathcal{M}$-progress-consistency of $\mathsf{GenReach}(\mathcal{L}(\mathcal{D}))$ can be decided in time $\mathcal{O}(|M|^2 \cdot |Q|^4 \cdot |C|)$ (notation $\mathcal{O}(\cdot)$ denotes the standard big O notation).

[FH10]: Fijalkow et al. (2010), *The surprizing complexity of reachability games*

[CFH14]: Colcombet et al. (2014), *Playing Safe*

Dilworth's theorem states that in a finite partially ordered set, the size of the smallest decomposition into chains matches the size of the largest antichain.

[Dil50]: Dilworth (1950), *A Decomposition Theorem for Partially Ordered Sets*

The smallest number of chains of a partial order can be computed in polynomial time [FRS03]; hence, adding requirement (b) moves the problem to a higher complexity class.

[FRS03]: Felsner et al. (2003), *Recognition Algorithms for Orders of Small Width and Graphs of Small Dilworth Number*

7.6 Additional proofs and missing technical details

We dedicate this section to the complete proofs of results from Chapter 7 whose proofs were omitted.

7.6.1 Technical details for general safety objectives

We prove here the characterization of the memory requirements of general safety objectives through $\mathcal{M}$-strong-monotony discussed in Section 7.3. We restate and prove Theorem 7.3.3.

Theorem 7.3.3 (Characterization for safety) *Let $W \subseteq C^\omega$ be a general safety objective and $\mathcal{M}$ be a memory structure.*

1. *If the prefix preorder $\preceq$ of W is well-founded (in particular, if W is regular), then $\mathcal{M}$ suffices to play optimally for $\mathcal{P}_1$ if and only if W is $\mathcal{M}$-strongly-monotone.*
2. *In the general case, $\mathcal{M}$ suffices to play optimally for $\mathcal{P}_1$ in all finitely branching arenas if and only if W is $\mathcal{M}$-strongly-monotone.*

Proof. Let $\mathcal{D}_W = (Q, C, q_{\text{init}}, \delta, F)$ be the (possibly infinite) prefix-classifier automaton of W, and let $\mathcal{M} = (M, m_{\text{init}}, \alpha_{\text{upd}})$ be a (possibly infinite) memory structure.

The necessity of $\mathcal{M}$-strong-monotony (in both cases) was proved in Proposition 7.3.1. We now prove the sufficiency of $\mathcal{M}$-strong-monotony. We assume that W is $\mathcal{M}$-strongly-monotone. We write w.l.o.g. $F = \{q_{\text{fin}}\}$. Let $\mathcal{A} = (V, V_1, V_2, E)$ be an arena. As per the hypotheses, we require that $\preceq$ is well-founded or that $\mathcal{A}$ is finitely branching.

> This is the only place where, for generality, we allow memory structures with an infinite state space, as it works with no additional arguments.

We want to build a strategy σ based on $\mathcal{M}$ and optimal for $\mathcal{P}_1$ in $\mathcal{A}$. The key to the proof is to understand the following sets of states of $\mathcal{D}$ in order to know what to play in each pair $(v, m) \in V_1 \times M$. For $v \in V$, $m \in M$, we define

$$Q_{v,m} = \{q \in \Gamma_m \mid \mathcal{P}_1 \text{ has a winning strategy for } q^{-1}W \text{ from } v\}.$$

States in $Q_{v,m}$ are states of $\mathcal{D}$ that could be reached while the memory state is m, by definition of Γ_m. Notice that $q_{\text{fin}} \notin Q_{v,m}$ for all v and m, as $\mathcal{P}_1$ cannot win for objective $q_{\text{fin}}^{-1}W = \emptyset$. The $\mathcal{M}$-strong-monotony hypothesis tells us that each Γ_m is a chain for $\preceq$ (Lemma 7.3.2), so each $Q_{v,m}$ is too.

> We recall that notation Γ_m was defined in Section 7.3 on page 159.

We define a strategy $\sigma: V_1 \times M \to E$ of $\mathcal{P}_1$ based on memory $\mathcal{M}$. Let $v \in V_1$, $m \in M$. We distinguish three cases.

▶ If $Q_{v,m}$ is empty, then it means that the game has reached a situation where it cannot be won anymore, so $\sigma(v, m)$ is chosen arbitrarily.

▶ Otherwise, if $Q_{v,m}$ has a minimum $q_{v,m}$ for $\preceq$, then this minimum represents the worst (for $\preceq$) state of Γ_m for which $\mathcal{P}_1$ still has a winning strategy. To play "safely", we play as if we wanted to win for this worst state. By definition of $Q_{v,m}$, there is a strategy $\sigma_{v,m}$ winning for $\mathcal{P}_1$ for $q_{v,m}^{-1}W$ from v. We define $\sigma(v, m) = \sigma_{v,m}(\lambda_v)$. Note that when $Q_{v,m}$ is non-empty, it always has a minimum if $\preceq$ is well-founded.

> We recall that λ_v is the empty history starting in v.

▶ If $Q_{v,m}$ is not empty and has no minimum, $\preceq$ is not well-founded; we fall in this case under the hypothesis that $\mathcal{A}$ is finitely branching. For $q \in Q_{v,m}$, let

$$E_{v,q} = \{\sigma'(\lambda_v) \in E \mid \sigma' \text{ is winning for } q^{-1}W \text{ from } v\}$$

be the set of outgoing edges of v that are taken immediately by at least one strategy winning for $q^{-1}W$ from v. We make three observations on sets $E_{v,q}$.

- For $q \in Q_{v,m}$, set $E_{v,q}$ is non-empty as $\mathcal{P}_1$ has a winning strategy for $q^{-1}W$ from v.
- For $q \in Q_{v,m}$, set $E_{v,q}$ is finite as v has finitely many outgoing edges.
- For $q, q' \in Q_{v,m}$, if $q \preceq q'$, then $E_{v,q} \subseteq E_{v,q'}$ as every strategy winning for $q^{-1}W$ is winning for $(q')^{-1}W$.

As sets $E_{v,q}$ are non-empty, finite, and non-decreasing (w.r.t. q), this means that the intersection $\bigcap_{q \in Q_{v,m}} E_{v,q}$ is non-empty. Let $e \in \bigcap_{q \in Q_{v,m}} E_{v,q}$; we define $\sigma(v, m) = e$.

We have now defined σ; we show that it is optimal. Let $v_0 \in V$ be such that $\mathcal{P}_1$ has a winning strategy for objective W from v_0. Let $\rho = e_1 e_2 \dots \in E^\omega$ be a play consistent with σ from v_0, and $w = \mathsf{col}^\omega(\rho)$. We write $w = c_1 c_2 \dots$ and we show that $w \in W$. As W is a general safety objective, this amounts to showing that for every finite prefix $w_i = c_1 \dots c_i$ of w, $\delta^*(q_{\mathsf{init}}, w_i) \neq q_{\mathsf{fin}}$. For $i \geq 0$, let $q_i = \delta^*(q_{\mathsf{init}}, w_i)$, $e_i = (v_{i-1}, c_i, v_i)$, and $m_i = \alpha^*_{\mathsf{upd}}(m_{\mathsf{init}}, w_i)$. We show by induction on i that for all $i \geq 0$, $q_i \in Q_{v_i, m_i}$. This suffices to prove the claim, as $q_{\mathsf{fin}} \notin Q_{v_i, m_i}$ for all $i \geq 0$.

For $i = 0$, we have $w_i = \varepsilon$, so $m_0 = \alpha^*_{\mathsf{upd}}(m_{\mathsf{init}}, w_i) = m_{\mathsf{init}}$ and $q_0 = \delta^*(q_{\mathsf{init}}, w_i) = q_{\mathsf{init}}$. By definition, we have $q_{\mathsf{init}} \in \Gamma_{m_{\mathsf{init}}}$. As $\mathcal{P}_1$ has a winning strategy for $W = q_{\mathsf{init}}^{-1} W$ from v_0 by hypothesis, we have that $q_0 \in Q_{v_0, m_0}$.

We now assume that $q_i \in Q_{v_i, m_i}$ for some $i \geq 0$. As $q_i \in \Gamma_{m_i}$, we have that $q_{i+1} = \delta(q_i, c_{i+1}) \in \Gamma_{\alpha_{\mathsf{upd}}(m_i, c_{i+1})} = \Gamma_{m_{i+1}}$. To show that $q_{i+1} \in Q_{v_{i+1}, m_{i+1}}$, it is left to show that there is a winning strategy for $q_{i+1}^{-1} W$ from v_{i+1}. We know that Q_{v_i, m_i} is not empty, and we distinguish three cases.

- ▶ If $v_i \in V_2$, then since $\mathcal{P}_1$ has a strategy winning for $q_i^{-1} W$ from v_i, $\mathcal{P}_1$ must be able to win no matter the choice of $\mathcal{P}_2$ in v_i. Hence, $\mathcal{P}_1$ has a winning strategy from $\delta(q_i, c_{i+1})^{-1} W = q_{i+1}^{-1} W$ from v_{i+1}.
- ▶ If $v_i \in V_1$ and Q_{v_i, m_i} has a minimum q_{v_i, m_i}, then e_{i+1} is consistent with a strategy σ_{v_i, m_i} winning for $q_{v_i, m_i}^{-1} W$ from v. This strategy also wins for $q_i^{-1} W$, as $q_{v_i, m_i} \leq q_i$. Thus, there must also be a strategy winning for $\delta(q_i, c_{i+1})^{-1} W = q_{i+1}^{-1} W$ from v_{i+1}.
- ▶ If $v_i \in V_1$ and Q_{v_i, m_i} has no minimum, then as $q_i \in Q_{v_i, m_i}$, there is in particular a winning strategy for $q_i^{-1} W$ from v_i that takes edge $\sigma(v_i, m_i) = (v_i, c_{i+1}, v_{i+1})$. Thus, $\mathcal{P}_1$ has a strategy winning for $\delta(q_i, c_{i+1})^{-1} W = q_{i+1}^{-1} W$ from v_{i+1}. □

7.6.2 Technical details for regular reachability objectives

In this section, we prove Theorem 7.4.8 discussed in Section 7.4, which characterizes the memory requirements of regular reachability objectives. In order to prove this characterization, we start with extra preliminaries on the notion of *tree induced by a strategy*, and a classical way to define a notion of *height* for these trees.

Let $\mathcal{D}$ be a (possibly infinite) automaton and $W = \mathsf{GenReach}(\mathcal{L}(\mathcal{D}))$ be the induced reachability objective. Let $\mathcal{A} = (V, V_1, V_2, E)$ be a (possibly infinite) arena. For $v \in V$ and σ a strategy of $\mathcal{P}_1$ on $\mathcal{A}$, we define $\mathcal{A}^{\sigma, v}$ to be the *tree induced by σ from v*, whose nodes are all the histories from v consistent with σ. It can be built by induction:

- ▶ it contains as a root the empty history λ_v from v;
- ▶ if γ is a history in $\mathcal{A}^{\sigma, v}$, then

- if $\mathsf{out}(\gamma) \in V_1$, γ has only one child which is $\gamma\sigma(\gamma)$;
- if $\mathsf{out}(\gamma) \in V_2$, γ has a child γe for all edges $e = (\mathsf{out}(\gamma), c, v') \in E$.

We denote $\mathcal{A}^{\sigma,v}_{|\mathcal{L}(\mathcal{D})}$ for the subtree of $\mathcal{A}^{\sigma,v}$ in which nodes γ whose projection to colors are a word in $\mathcal{L}(\mathcal{D})$ are not prolonged. A tree is called *well-founded* if it has no infinite branch. Notice that σ is winning from v if and only if $\mathcal{A}^{\sigma,v}_{|\mathcal{L}(\mathcal{D})}$ is well-founded. In a well-founded tree, we can associate an ordinal *rank* with each node (a generalization of the *height* for finite trees). By induction, for a leaf γ of the tree, we define $\mathsf{rank}(\gamma) = 0$, and for an internal node γ, we define $\mathsf{rank}(\gamma) = \sup\{\mathsf{rank}(\gamma') + 1 \mid \gamma' \text{ a child of } \gamma\}$. The rank of a tree is the rank of its root. More details on this notion of rank for well-founded relations can be found in [Kec95, Appendix B].

The rank of a well-founded tree with finite branching is necessarily $< \omega$; we use greater ordinals only when the trees have infinite branching. The upcoming proof works on arenas with arbitrary branching, but for (even infinite) arenas with *finite* branching, only finite trees with finite ranks are needed.

We now restate and prove Theorem 7.4.8.

> **Theorem 7.4.8** (Characterization for regular reachability) *Let $W \subseteq C^\omega$ be a* regular *reachability objective and $\mathcal{M}$ be a memory structure. Memory $\mathcal{M}$ suffices to play optimally for $\mathcal{P}_1$ if and only if W is $\mathcal{M}$-strongly-monotone and $\mathcal{M}$-progress-consistent.*

Proof. The necessity of the two conditions was proved respectively in Propositions 7.3.1 and 7.4.4.

We prove the sufficiency of the two conditions. Let $\mathcal{D}_W = (Q, C, q_{\mathsf{init}}, \delta, F)$ be the prefix-classifier automaton of W (which is finite as W is regular), and $\mathcal{M} = (M, m_{\mathsf{init}}, \alpha_{\mathsf{upd}})$. We write w.l.o.g. $F = \{q_{\mathsf{fin}}\}$. We assume that W is $\mathcal{M}$-strongly-monotone and $\mathcal{M}$-progress-consistent. Let $\mathcal{A} = (V, V_1, V_2, E)$ be a (possibly infinite) arena. We construct an optimal strategy based on memory $\mathcal{M}$, using the same idea as in the proof for safety objectives (Theorem 7.3.3): we once again consider a strategy based on memory $\mathcal{M}$ making choices that are "locally optimal", and deduce thanks to our hypotheses ($\mathcal{M}$-strong-monotony and $\mathcal{M}$-progress-consistency) that this strategy must be optimal.

For $v \in V$, $m \in M$, we define

$$q_{v,m} = \min_{\le}\{q \in \Gamma_m \mid$$

$$\mathcal{P}_1 \text{ has a winning strategy for objective } q^{-1}W \text{ from } v\},$$

or we fix $q_{v,m} = q_{\mathsf{fin}}$ if the set is empty (this is reasonable as q_{fin} is the greatest state for $\le$, and all strategies are winning for objective $q_{\mathsf{fin}}^{-1}W = C^\omega$). Notice that we rely on $\mathcal{M}$-strong-monotony and on regularity of W in this definition, as we are guaranteed that the min exists because Γ_m is a chain

and because Q is finite. For $v \in V_1$, $m \in M$, we also fix a strategy $\sigma_{v,m}$ of $\mathcal{P}_1$ that is winning for $q_{v,m}^{-1} W$ from v.

Let us take a step back. Like in the proof for safety, we want to define $\sigma(v, m)$ as the first edge taken by $q_{v,m}$ from v — we play locally reasonable edges played by good strategies and hope that this creates a "globally" optimal strategy. However, this does not work in general, as any choice for the strategies $\sigma_{v,m}$ may not be good: indeed, such strategies may be winning, but may make unnecessary moves delaying the achievement of the objective. For instance, in the arena of Figure 7.5, a strategy inducing bab^ω is winning for $q_{\text{init}}^{-1} W$, but not as fast as possible (it takes three moves to create a word in $\mathcal{L}(\mathcal{D})$, while it is possible to do it in two moves). If, by imitating the first move of this strategy, we define $\sigma(v, m_1) = (v, b, v)$, we then get stuck and σ plays the losing word b^ω.

Therefore, we make one additional requirement on $\sigma_{v,m}$: we assume that it is a strategy guaranteeing the *quickest win* from v for objective $q_{v,m}^{-1} W$. In other words, we take $\sigma_{v,m}$ such that the tree $\mathcal{A}_{|\mathcal{L}(\mathcal{D}_W)}^{\sigma_{v,m}, v}$ has the least ordinal rank $\theta_{v,m}$ among all winning strategies.

We define a strategy $\sigma \colon V_1 \times M \to E$ of $\mathcal{P}_1$ based on memory $\mathcal{M}$: for $v \in V_1$, $m \in M$, we set $\sigma(v, m) = \sigma_{v,m}(\lambda_v)$. We show that σ is optimal.

Let $v_0 \in V$ be a vertex from which $\mathcal{P}_1$ has a winning strategy for objective W. We show that σ wins from v_0. Let $\rho = (v_0, c_1, v_1)(v_1, c_2, v_2) \ldots \in E^\omega$ be a play consistent with σ from v_0, and $w = c_1 c_2 \ldots \in C^\omega$. For $i \geq 0$, we fix $m_i = \alpha_{\text{upd}}^*(m_{\text{init}}, c_1 \ldots c_i)$ and $q_i = \delta^*(q_{\text{init}}, c_1 \ldots c_i)$. We show that $w \in W$, i.e., that there exists $i \geq 0$ such that $q_i = q_{\text{fin}}$. For brevity, we also write $q_i' = q_{v_i, m_i}$ and $\theta_i = \theta_{v_i, m_i}$.

As there are finitely many memory states and finitely many automaton states, we can find $m \in M$, $q, q' \in Q$, and an infinite increasing sequence of indices $(i_j)_{j \geq 0}$ such that for all $j \geq 0$, $m_{i_j} = m$, $q_{i_j} = q$, and $q_{i_j}' = q'$. We decompose w into infinitely many finite words cut at every index i_j: for $j \geq 0$, let $w_j = c_{i_j+1} \ldots c_{i_{j+1}}$. If $q = q_{\text{fin}}$, we are done, as w indeed reaches the final state of $\mathcal{D}$. We now assume by contradiction that $q \neq q_{\text{fin}}$. As q is reached infinitely many times and q_{fin} is absorbing, this implies that $q_i \neq q_{\text{fin}}$ for all $i \geq 0$. We prove a few properties about the various sequences that we have defined.

This is where we use finiteness of $\mathcal{D}_W$, which must be circumvented to extend the proof to general — and not only regular — reachability objectives. However, the approach seems more fundamentally lacking for general reachability objectives, which is argued in Example 7.6.1 below.

(a) We first show that

$$\forall i \geq 0, \forall j \geq i, q_j' \preceq \delta(q_i', c_{i+1} \ldots c_j). \tag{7.2}$$

To do so, we show that for all $i \geq 0$, $q_{i+1}' \preceq \delta(q_i', c_{i+1})$, and Equation (7.2) then follows by induction. Let $i \geq 0$. As $\mathcal{P}_1$ has a winning strategy for $(q_i')^{-1} W$ from v_i, and playing (v_i, c_{i+1}, v_{i+1}) is an action consistent with winning strategy σ_{v_i, m_i}, $\mathcal{P}_1$ also has a winning strategy for $\delta(q_i', c_{i+1})^{-1} W$ from v_{i+1}. Moreover, as $q_i' \in \Gamma_{m_i}$, we have that

$\delta(q'_i, c_{i+1}) \in \Gamma_{\alpha_{\text{upd}}(m_i, c_{i+1})} = \Gamma_{m_{i+1}}$. Hence, $q'_{i+1} \leq \delta(q'_i, c_{i+1})$ as q'_{i+1} is defined as the minimum of a set in which $\delta(q'_i, c_{i+1})$ lies.

(b) We use this to show that the sequence $(q'_i)_{i \geq 0}$, which only depends on the arena vertices and the memory states visited, underapproximates the sequence $(q_i)_{i \geq 0}$, which corresponds to the actual automaton states visited by word w. Formally,

$$\forall i \geq 0, q'_i \leq q_i. \tag{7.3}$$

We prove it by induction. For $i = 0$, we have $q_0 = q_{\text{init}}$, and by hypothesis, $\mathcal{P}_1$ has a winning strategy from v_0 for objective $W = q_{\text{init}}^{-1} W$. Moreover, $m_0 = m_{\text{init}}$ and $q_{\text{init}} \in \Gamma_{m_{\text{init}}}$, so by the definition of minimum, $q'_0 \leq q_0$. We now assume that $q'_i \leq q_i$ for some $i \geq 0$. By Equation (7.2), we know that $q'_{i+1} \leq \delta(q'_i, c_{i+1})$. By Lemma 2.8.10, we have $\delta(q'_i, c_{i+1}) \leq \delta(q_i, c_{i+1}) = q_{i+1}$. We conclude that $q'_{i+1} \leq q_{i+1}$, which proves the claim. For all $i \geq 0$, as $q_i \neq q_{\text{fin}}$, we deduce moreover that $q'_i \neq q_{\text{fin}}$.

(c) We now prove that

$$\forall i \geq 0, q'_{i+1} = \delta(q'_i, c_{i+1}) \Rightarrow \theta_{i+1} < \theta_i. \tag{7.4}$$

Let $i \geq 0$ such that $q'_{i+1} = \delta(q'_i, c_{i+1})$. We know that the tree $\mathcal{A}^{\sigma_{v_i, m_i}, v_i, v}_{|\mathcal{L}(\mathcal{D}_W)}$ has rank θ_i. As $q'_i \neq q_{\text{fin}}$, $\theta_i \neq 0$. Hence, since playing (v_i, c_{i+1}, v_{i+1}) is consistent with strategy σ_{v_i, m_i}, it is possible to find a strategy that induces a tree from v_{i+1} for objective $\delta(q'_i, c_{i+1})^{-1} W$ of height strictly smaller than θ_i: we simply consider the strategy of the subtree of $\mathcal{A}^{\sigma_{v_i, m_i}, v_i, v}_{|\mathcal{L}(\mathcal{D}_W)}$ with root (v_i, c_{i+1}, v_{i+1}). As $\delta(q'_i, c_{i+1}) = q'_{i+1}$ by hypothesis, we deduce that there is a strategy that wins for objective $(q'_{i+1})^{-1} W$ from v_{i+1} and whose tree has height $< \theta_i$. We conclude that $\theta_{i+1} < \theta_i$.

(d) We show a final property:

$$\forall i \geq i_0, q'_{i+1} = \delta(q'_i, c_{i+1}). \tag{7.5}$$

By Equation (7.2), the only other option, which we assume by contradiction, is that there is $k \geq i_0$ such that $q'_{k+1} < \delta(q'_k, c_{k+1})$. Let $j \geq 0$ such that $i_j \leq k < i_{j+1}$. We split w_j into two parts: $w_j^{(1)} = c_{i_j+1} \ldots c_{k+1}$ and $w_j^{(2)} = c_{k+2} \ldots c_{i_{j+1}}$. First, notice that $q'_{k+1} < \delta^*(q'_{i_j}, w_j^{(1)})$. Indeed, $q'_k \leq \delta^*(q'_{i_j}, c_{i_j+1} \ldots c_k)$ by Equation (7.2), and $q'_{k+1} < \delta(q'_k, c_{k+1})$ by hypothesis. Secondly, we have that $q'_{i_{j+1}} \leq \delta^*(q'_{k+1}, w_j^{(2)})$ by Equation (7.2). We recall that $q'_{i_j} = q'_{i_{j+1}} = q'$. We deduce that

$$\begin{aligned} q'_{k+1} &< \delta^*(q'_{i_j}, w_j^{(1)}) \\ &= \delta^*(q'_{i_{j+1}}, w_j^{(1)}) \\ &\leq \delta^*(\delta^*(q'_{k+1}, w_j^{(2)}), w_j^{(1)}) \\ &= \delta^*(q'_{k+1}, w_j^{(2)} w_j^{(1)}). \end{aligned}$$

We therefore have that $w_j^{(2)} w_j^{(1)}$ makes progress from q'_{k+1}. As $w_j = w_j^{(1)} w_j^{(2)}$ is a cycle on memory state m, we have that $w_j^{(2)} w_j^{(1)}$ must be a cycle on memory state $m_k = \alpha^*_{\mathsf{upd}}(m, w_j^{(1)})$. By $\mathcal{M}$-progress-consistency, this means that $(w_j^{(2)} w_j^{(1)})^\omega \in (q'_{k+1})^{-1} W$, so $(w_j^{(1)} w_j^{(2)})^\omega = (w_j)^\omega \in (q'_{i_j})^{-1} W$. By Equation (7.3), this implies that $(w_j)^\omega \in q_{i_j}^{-1} W$. However,

$$\delta^*(q_{i_j}, w_j) = q_{i_j} \neq q_{\mathsf{fin}},$$

so repeating w_j from q_{i_j} cannot be winning. This is a contradiction, which means that Equation (7.5) holds.

We now use Equation (7.4) and Equation (7.5) to draw a contradiction with our initial hypothesis that $q \neq q_{\mathsf{fin}}$. For every index $i \geq i_0$ onward, we have that $q'_{i+1} = \delta(q'_i, c_{i+1})$ (Equation (7.5)). By Equation (7.4), this means that the infinite ordinal sequence $(\theta_i)_{i \geq i_0}$ is decreasing, which is impossible. $\qquad\square$

We do not know whether a generalization to general reachability objectives played on infinite arenas (i.e., Conjecture 7.4.7) holds. Using the one-to-two-player lift from Chapter 4, as we did in Subsection 7.4.3, will not help as it only deals with games played on finite arenas. We provide an example showing that our proof technique above for Theorem 7.4.8 fails for some general reachability objective with well-founded preorder.

Example 7.6.1 Let $C = \mathbb{N}$. We define a general reachability objective

$$W = \{c_1 c_2 \ldots \in C^\omega \mid \exists i < j, c_i \geq c_j\}$$

consisting of all the infinite sequences that are not increasing. We represent its (infinite) prefix-classifier automaton $\mathcal{D}_W$ in Figure 7.9. For preorder $\leq$, we have that $q_{\mathsf{init}} < q_i < q_{\mathsf{fin}}$ for all $i \geq 0$, and $q_i \leq q_j$ if and only if $i \leq j$. We observe that

▶ W is $\mathcal{M}_{\mathsf{triv}}$-strongly-monotone as preorder $\leq$ is total;
▶ W is $\mathcal{M}_{\mathsf{triv}}$-progress-consistent as repeating any color is immediately winning.

Moreover, $\leq$ is well-founded as every set of states of $\mathcal{D}_W$ has a minimum, so Remark 7.3.6 does not apply. If Conjecture 7.4.7 indeed holds, then $\mathcal{M}_{\mathsf{triv}}$ suffices here. Unfortunately, our proof technique for Theorem 7.4.8 does not work here. Let $\mathcal{A}$ be the finitely branching arena in Figure 7.9. There is a winning strategy from every state. Referencing the vocabulary of the proof of Theorem 7.4.8, the strategy guaranteeing the quickest win from a vertex v_i is the strategy starting with $(v_i, i, v_{i+1})(v_{i+1}, i, v_{i+2})$, which wins in two moves. This means that strategy σ built in the proof of Theorem 7.4.8 plays (v_i, i, v_{i+1}) in v_i. But the infinite play generated by σ from v_0 then sees colors $0, 1, 2, 3, \ldots$, which is not a winning word.
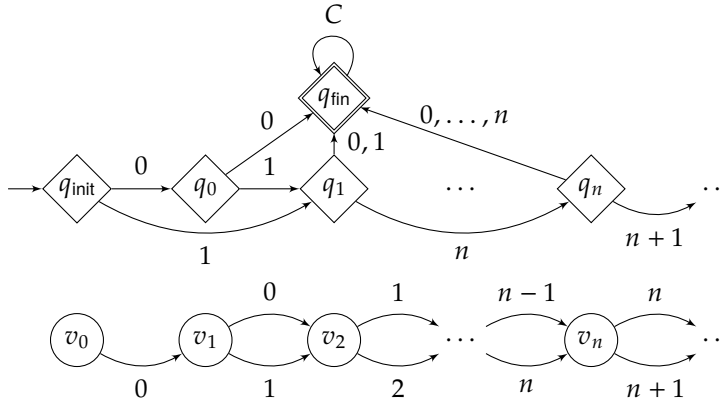
7.6.3 Technical details for computational complexity

In Lemma 7.5.3, we have rephrased the existence of a memory structure $\mathcal{M}$ with k states such that a general reachability or safety objective is $\mathcal{M}$-strongly-monotone into the existence of a *monotone decomposition* with k sets. We therefore consider the following decision problem.

The definition of *monotone decomposition* is in Definition 7.5.2 on page 173.

MONOTONY

Input: A DFA $\mathcal{D}$ and an integer $k \in \mathbb{N}$.
Question: Is there a monotone decomposition of $\mathcal{D}$ with at most k sets?

As per Lemma 7.5.3, this problem is equivalent to asking whether there is a memory structure $\mathcal{M}$ with at most k states such that a regular objective W derived from $\mathcal{D}$ is $\mathcal{M}$-strongly-monotone (Lemma 7.5.3), or whether there is a chromatic memory structure with $\leq k$ states that suffices to play optimally for $\mathcal{P}_1$ for $\mathsf{GenSafe}(\mathcal{L}(\mathcal{D}))$ (Theorem 7.3.3). It is also related, though not equivalent, to the chromatic memory requirements of $\mathsf{GenReach}(\mathcal{L}(\mathcal{D}))$ (Theorem 7.4.8). We will show that the MONOTONY problem is NP-complete.

Membership in NP. We discuss here that the decision problems related to the properties used in our characterizations of chromatic memory requirements, $\mathcal{M}$-strong-monotony and $\mathcal{M}$-progress-consistency, are in NP. The idea is simply that, given a DFA $\mathcal{D}$ and a memory structure $\mathcal{M}$, we can decide in polynomial time whether the objectives derived from $\mathcal{D}$ are $\mathcal{M}$-strongly-monotone, and whether they are $\mathcal{M}$-progress-consistent.

> **Lemma 7.6.2** MEMORY-SAFE *is in* NP. *Given an input DFA* $\mathcal{D} = (Q, C, q_{\mathsf{init}}, \delta, F)$ *and a witness decomposition with* $k \leq |Q|$ *sets, we can check whether it forms a* monotone *decomposition in* $\mathcal{O}(|Q|^4 \cdot |C|)$.

Proof. We show that the MONOTONY problem belongs to NP, which is equivalent to our statement thanks to Lemma 7.5.3. Let $\mathcal{D} = (Q, C, q_{\mathsf{init}}, \delta, F)$ be a DFA and $k \in \mathbb{N}$. Notice that if $k \geq |Q|$, the answer to the problem

is always YES, as $(\{q\})_{q \in Q}$ is always a monotone decomposition. It is left to consider the case $k < |Q|$. A monotone decomposition with k sets of states of $\mathcal{D}$ therefore has size polynomial in the inputs. We can verify that such sets indeed form a monotone decomposition in polynomial time, by checking each of the three requirements from the definition of monotone decomposition.

▶ For requirement (a) (the decomposition covers the state space), we simply need to iterate on all the states appearing in the monotone decomposition, which has time complexity $\mathcal{O}(k \cdot |Q|)$.

▶ For requirement (b) (the decomposition is "stable" by reading colors), we simply have to check that for each set and each color, the image of the set by this color is included in another set, which can be done naively in $\mathcal{O}(k^2 \cdot |C| \cdot |Q|)$.

▶ For requirement (c) (each set is a chain), we discuss how to check in polynomial time that each set is a chain. One way to do it is to precompute, for every pair $q_1, q_2 \in Q$, whether $q_1 \preceq q_2$, $q_2 \preceq q_1$, or none of these. This amounts to solving *language containment queries*, which can be done in polynomial time for regular languages recognized by deterministic finite automata given as an input. Computing all these relations can be done in $\mathcal{O}(|Q|^2 \cdot (|Q|^2 \cdot |C|)) = \mathcal{O}(|Q|^4 \cdot |C|)$ as there are $|Q|^2$ pairs of states to consider, and each containment query can be performed in $\mathcal{O}(|Q|^2 \cdot |C|)$. Once all these relations have been precomputed, checking whether each set of the decomposition is a chain can be done in linear time, as a chain is simply a set in which all pairs of elements are comparable.

As $k \leq |Q|$, the overall time complexity is $\mathcal{O}(|Q|^4 \cdot |C|)$. □

The classical procedure for language containment queries is as follows: observe that $\mathcal{L}(\mathcal{D}_1) \subseteq \mathcal{L}(\mathcal{D}_2)$ if and only if $\mathcal{L}(\mathcal{D}_1) \cap \overline{\mathcal{L}(\mathcal{D}_2)} = \emptyset$. Deterministic finite automata can be complemented by complementing their final states, and emptiness of the intersection can be checked by building the direct product of the automata. The product construction was first formalized in [RS59].

[RS59]: Rabin et al. (1959), *Finite Automata and Their Decision Problems*

For regular reachability objectives, we express the notion of $\mathcal{M}$-progress-consistency in a way that makes decidability in polynomial time clear.

> **Lemma 7.6.3** *Let* $\mathcal{D} = (Q, C, q_{\mathsf{init}}, \delta, F)$ *be a DFA,* $W = \mathsf{GenReach}(\mathcal{L}(\mathcal{D}))$ *be the derived regular reachability objective, and* $\mathcal{M} = (M, m_{\mathsf{init}}, \alpha_{\mathsf{upd}})$ *be a memory structure. We assume w.l.o.g. that* $\mathcal{D}$ *has a single final state* q_{fin} *which is absorbing.*
> *Objective* W *is* $\mathcal{M}$*-progress-consistent if and only if for all* $m \in M$, $q_1 \in Q$,
>
> $(L_{m_{\mathsf{init}}, m} \cap L_{q_{\mathsf{init}}, q_1} \neq \emptyset) \Longrightarrow$
> $\qquad (\forall q_2 \in Q \text{ s.t. } q_2 \neq q_{\mathsf{fin}} \text{ and } q_1 \prec q_2, \ L_{m,m} \cap L_{q_1,q_2} \cap L_{q_2,q_2} = \emptyset).$

This result reduces the search for words that witness "non-$\mathcal{M}$-progress-consistency" to a more constrained situation. In general, if a word $w \in L_{m,m}$ witnesses that W is not $\mathcal{M}$-progress-consistent because it makes progress from a state q but does not win when repeated from q, then we may have to read w multiple times on the automaton before noticing that repeating it does not reach q_{fin}. However, in such a situation, we can actually find two states $q_1 \prec q_2$ such that w is read from q_1 to q_2 and w is a cycle on q_2

— in other words, just by reading w twice on the right state, we can notice that w contradicts $\mathcal{M}$-progress-consistency.

Proof. The left-to-right implication can be shown by contrapositive. Negating the implication gives a $w_1 \in L_{m_{\text{init}},m} \cap L_{q_{\text{init}},q_1}$ and a $w_2 \in L_{m,m} \cap L_{q_1,q_2} \cap L_{q_2,q_2}$ such that $w_1 \prec w_1 w_2$ and $w_1(w_2)^\omega$ does not go through q_{fin}, so $w_1(w_2)^\omega \notin W$. This shows that W is not $\mathcal{M}$-progress-consistent.

For the right-to-left implication, we assume by contrapositive that W is not $\mathcal{M}$-progress-consistent: there exist $m \in M$, $w_1 \in L_{m_{\text{init}},m}$, $w_2 \in L_{m,m}$ such that $w_1 \prec w_1 w_2$ and $w_1(w_2)^\omega \notin W$. For $i \geq 0$, let $q_i' = \delta^*(q_{\text{init}}, w_1(w_2)^i)$. We have $q_0' \prec q_1'$ since $w_1 \prec w_1 w_2$. By induction and by Lemma 2.8.10, the sequence $(q_i')_{i \geq 0}$ is non-decreasing. As there are finitely many states, it therefore reaches a fixpoint, which cannot be q_{fin} as $w_1(w_2)^\omega \notin W$. We denote $q_2 = q_j'$ its fixpoint and $q_1 = q_{j-1}'$ the last state before reaching the fixpoint (in particular, $q_1 \prec q_2$, $\delta^*(q_1, w_2) = q_2$, and $\delta^*(q_2, w) = q_2$).

We have that $w_1(w_2)^{j-1} \in L_{m_{\text{init}},m} \cap L_{q_{\text{init}},q_1}$, $q_1 \prec q_2$, $q_2 \neq q_{\text{fin}}$, and $w_2 \in L_{m,m} \cap L_{q_1,q_2} \cap L_{q_2,q_2}$, which shows that we do not have the implication from the statement. $\qquad\square$

This condition is easy to check algorithmically, as it consists of checking emptiness and non-emptiness of intersections of regular languages for all memory states m and all pairs q_1, q_2 of comparable states of $\mathcal{D}$.

> **Corollary 7.6.4** MEMORY-REACH *is in* NP. *Given an input DFA* $\mathcal{D} = (Q, C, q_{\text{init}}, \delta, F)$ *and a witness memory structure* $\mathcal{M} = (M, m_{\text{init}}, \alpha_{\text{upd}})$ *with* $|M| \leq |Q|$, *we can check whether* $\mathcal{M}$ *suffices to play optimally for* GenReach($\mathcal{L}(\mathcal{D})$) *in* $\mathcal{O}(|M|^2 \cdot |Q|^4 \cdot |C|)$.

Proof. Let $\mathcal{D} = (Q, C, q_{\text{init}}, \delta, F)$ be a DFA and $k \in \mathbb{N}$. If $k \geq |Q|$, then the answer to MEMORY-REACH($\mathcal{D}, k$) is always YES, as using $\mathcal{D}$ as a memory structure always suffices to play optimally. Indeed, $\mathcal{D}$-strong-monotony and $\mathcal{D}$-progress-consistency of an objective induced by $\mathcal{D}$ trivially hold (for $\mathcal{D}$-strong-monotony, by using $\mathcal{D}$ as a memory structure, we always know precisely the current class of prefixes, which is even stronger than knowing a chain; for $\mathcal{D}$-progress-consistency, any progress necessarily changes the state as two words with distinct winning continuations cannot reach the same state of $\mathcal{D}$). It is left to consider the case $k < |Q|$. A sufficient memory structure $\mathcal{M}$ with k states then has size polynomial in the inputs. To check that it suffices to play optimally, we need to verify that W is $\mathcal{M}$-strongly-monotone and $\mathcal{M}$-progress-consistent.

From memory $\mathcal{M}$, we can build the sets Γ_m for each state m of $\mathcal{M}$ by computing the product of $\mathcal{D}$ and $\mathcal{M}$ in $\mathcal{O}(|M| \cdot |Q| \cdot |C|)$. We can then check that family $(\Gamma_m)_{m \in M}$ forms a monotone decomposition in time $\mathcal{O}(|Q|^4 \cdot |C|)$ (Lemma 7.6.2). This means that $\mathcal{M}$-strong-monotony can be checked in polynomial time.

The $\mathcal{M}$-progress-consistency property can also be checked in polynomial time using Lemma 7.6.3. We first compute all pairs of states (q_1, q_2) such that $q_1 \prec q_2$, which was already discussed in Lemma 7.6.2 and can be done in $\mathcal{O}(|Q|^4 \cdot |C|)$. Then, we can find every triplet $(m, q_1, q_2) \in M \times Q^2$ such that $L_{m_{\text{init}}, m} \cap L_{q_{\text{init}}, q_1} \neq \emptyset$, $q_2 \neq q_{\text{fin}}$, and $q_1 \prec q_2$. For each of these triplets, we need to check the emptiness of language $L_{m,m} \cap L_{q_1, q_2} \cap L_{q_2, q_2}$, which can be represented by a product involving $\mathcal{M}$ and two copies of $\mathcal{D}$ (which has $|M| \cdot |Q|^2 \cdot |C|$ transitions). The time complexity of this step is then $\mathcal{O}((|M| \cdot |Q|^2) \cdot (|M| \cdot |Q|^2 \cdot |C|)) = \mathcal{O}(|M|^2 \cdot |Q|^4 \cdot |C|)$.

The overall procedure therefore has complexity $\mathcal{O}(|M|^2 \cdot |Q|^4 \cdot |C|)$. $\quad\square$

NP-hardness. We show that the Monotony problem is NP-hard, using a reduction from the (directed) HamiltonianCycle problem, which is NP-complete [Kar72]. In the following, a *(directed) graph* is a tuple $G = (V, E)$ with $E \subseteq V \times V$. A *Hamiltonian cycle of G* is a sequence $(u_1, \ldots, u_n)$ in which each vertex of V appears exactly once, $(u_i, u_{i+1}) \in E$ for all i, $1 \le i < n$, and $(u_n, u_1) \in E$.

[Kar72]: Karp (1972), *Reducibility Among Combinatorial Problems*
Edges in the set denoted E are, for this one occasion, not colored.

HamiltonianCycle
Input: A directed graph $G = (V, E)$.
Question: Is there a *Hamiltonian cycle* in G?

> **Proposition 7.6.5** Monotony *is NP-hard. More precisely, for every graph $G = (V, E)$, there is a polynomial-size DFA $\mathcal{D}_G$ such that G has a Hamiltonian cycle if and only if $\mathcal{D}_G$ has a monotone decomposition with $|V| + |E| + 1$ sets. Objective GenReach$(\mathcal{L}(\mathcal{D}_G))$ is moreover $\mathcal{M}_{\text{triv}}$-progress-consistent.*

Proof. We start by defining an operator Automaton($\cdot$) turning a directed graph into a DFA. Let $G = (V, E)$ be a directed graph. We define Automaton(G) as the DFA $(Q, \Sigma, \delta, q_{\text{init}}, F)$ with $Q = V \uplus E$, $\Sigma = \{\text{in}, \text{out}\}$, and transitions such that

- for $v \in V$, $\delta(v, \text{in}) = \delta(v, \text{out}) = v$;
- for $e = (v_1, v_2) \in E$, $\delta(e, \text{in}) = v_1$ and $\delta(e, \text{out}) = v_2$.

We ignore q_{init} and F at the moment. This definition is inspired from a reduction in [Boo78] (although the rest of the proof is different).

[Boo78]: Booth (1978), *Isomorphism Testing for Graphs, Semigroups, and Finite Automata Are Polynomially Equivalent Problems*

Let us consider a graph $G = (V, E)$ as an input to the HamiltonianCycle problem. We show how to transform it in a polynomial-size DFA for which the answer to the Monotony problem (along with a well-chosen $k \in \mathbb{N}$) corresponds. We illustrate this construction in Figure 7.10. Let $n = |V|$ and $m = |E|$. We assume that $m \ge n$ (otherwise, G cannot have a Hamiltonian cycle). We also consider the *cycle graph with n vertices* $C_n = (V_C, E_C)$, with $V_C = \{v_1^C, \ldots, v_n^C\}$ and $E_C = \{e_1^C, \ldots, e_n^C\}$ such that $e_i^C = (v_i^C, v_{i+1}^C)$ for $1 \le i < n$ and $e_n = (v_n^C, v_1^C)$. We now consider a DFA $\mathcal{D}_G = (Q, \Sigma, \delta, q_{\text{init}}, F)$
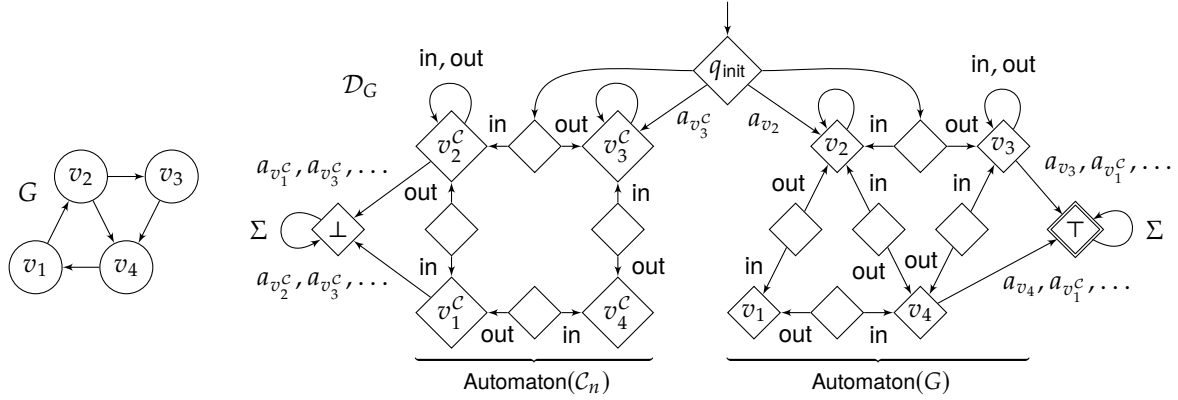
Figure 7.10: Illustration of DFA $\mathcal{D}_G$ starting from a graph G with four vertices. Only a few transitions of each kind are shown. Kinds of transitions that are not completely represented include: transitions from q_{init} to all states $z \in (V_\mathcal{C} \cup E_\mathcal{C}) \cup (V \cup E)$ with letter a_z, self-loops on all states in $V_\mathcal{C} \cup V$ with letters in and out, transitions from all states $z \in (V_\mathcal{C} \cup E_\mathcal{C}) \cup (V \cup E)$ to $\top$ with letter a_z, transitions from all states in V to $\top$ with letters $a_{v^\mathcal{C}}$ with $v^\mathcal{C} \in V_\mathcal{C}$, transitions from all states in E to $\top$ with letters $a_{e^\mathcal{C}}$ with $e^\mathcal{C} \in E_\mathcal{C}$, and transitions from all states in $(V_\mathcal{C} \cup E_\mathcal{C}) \cup (V \cup E)$ to $\bot$ for letters a_z that do not go to $\top$.

based on the disjoint union $\text{Automaton}(\mathcal{C}_n) \uplus \text{Automaton}(G)$ along with one new letter for each state and three extra states:

▶ $Q = (V_\mathcal{C} \uplus E_\mathcal{C}) \uplus (V \uplus E) \uplus \{q_{\text{init}}, \bot, \top\}$,
▶ $\Sigma = \{\text{in}, \text{out}\} \uplus \{a_z \mid z \in (V_\mathcal{C} \cup E_\mathcal{C}) \cup (V \cup E)\}$,
▶ $F = \{\top\}$.

The transitions with in and out are defined as above for states of $(V_\mathcal{C} \cup E_\mathcal{C}) \cup (V \cup E)$, and are self-loops on $q_{\text{init}}, \bot$ and $\top$.

The sole purpose of the new letters a_z is to induce a relevant ordering $\preceq$ — intuitively, we want $\bot$ to be the smallest state, $\top$ to be the largest, and all automaton states corresponding to vertices (resp. edges) of $\text{Automaton}(\mathcal{C}_n)$ to be smaller than all automaton states corresponding to vertices (resp. edges) of $\text{Automaton}(G)$, while making all other pairs of states non-comparable. Formally, for $z, z' \in (V_\mathcal{C} \cup E_\mathcal{C}) \cup (V \cup E)$ we define

$$\delta(z, a_{z'}) = \begin{cases} \top & \text{if } z = z', \\ \top & \text{if } z \in V \text{ and } z' \in V_\mathcal{C}, \\ \top & \text{if } z \in E \text{ and } z' \in E_\mathcal{C}, \\ \bot & \text{otherwise.} \end{cases}$$

We moreover define, for all $z \in (V_\mathcal{C} \cup E_\mathcal{C}) \cup (V \cup E)$, $\delta(q_{\text{init}}, a_z) = z$, $\delta(\bot, a_z) = \bot$, and $\delta(\top, a_z) = \top$.

We sum up the relations between the elements that follow from this construction:

▶ for all $q \in Q \setminus \{\bot\}$, $\bot \prec q$,
▶ for all $q \in Q \setminus \{\top\}$, $q \prec \top$,
▶ for all $v^\mathcal{C} \in V_\mathcal{C}$, for all $v \in V$, $v^\mathcal{C} \prec v$,
▶ for all $e^\mathcal{C} \in E_\mathcal{C}$, for all $e \in E$, $e^\mathcal{C} \prec e$,
▶ all other pairs of distinct states are non-comparable for $\preceq$.

The largest antichain in $\mathcal{D}_G$ for $\leq$ is attained by $V \cup E \cup \{q_{\text{init}}\}$: all these states are non-comparable, and all other states are comparable to some of them. This antichain has cardinality $n + m + 1$. Therefore, a monotone decomposition of $\mathcal{D}_G$ has size at least $n + m + 1$. We claim that it can have a size of exactly $n + m + 1$ if and only if G has a Hamiltonian cycle. This suffices to end the proof, since the size of $\mathcal{D}_G$ is polynomial in the size of G, and HAMILTONIANCYCLE(G) then returns YES if and only if MONOTONE($\mathcal{D}_G, n + m + 1$) returns YES.

> **Claim** *Graph G has a Hamiltonian cycle if and only if $\mathcal{D}_G$ has a monotone decomposition with $n + m + 1$ sets.*

We first prove the left-to-right implication. We assume that G has a Hamiltonian cycle $(u_1, \ldots, u_n)$. Let $e_i = (u_i, u_{i+1})$ for $1 \leq i < n$, and $e_n = (u_n, u_1)$. Let $E \setminus \{e_1, \ldots, e_n\} = \{e_{n+1}, \ldots, e_m\}$. The fact that there is a Hamiltonian cycle in G allows for a natural pairing of vertices (resp. edges) of $\mathcal{C}_n$ with vertices (resp. edges) of G in sets of a monotone decomposition, which will in particular be closed under reading in and out. We define sets $(\Gamma_i)_{1 \leq i \leq n+m+1}$ such that:

► for $1 \leq i \leq n$, $\Gamma_i = \{\bot, v_i^{\mathcal{C}}, u_i, \top\}$;
► for $1 \leq i \leq n$, $\Gamma_{n+i} = \{\bot, e_i^{\mathcal{C}}, e_i, \top\}$;
► for $1 \leq i \leq m - n$, $\Gamma_{2n+i} = \{\bot, e_{n+i}, \top\}$;
► $\Gamma_{n+m+1} = \{\bot, q_{\text{init}}, \top\}$.

We check that these sets form a monotone decomposition of $\mathcal{D}_G$. These sets cover the states of $\mathcal{D}_G$, and they are chains by construction. It is left to check the second requirement (b) of a monotone decomposition. Let $i \in \{1, \ldots, n+m+1\}$. If $i \geq 2n+1$, then Γ_i has three elements. For $a \in \Sigma$, the set $\delta(\Gamma_i, a)$ is a set with at most three elements that includes $\bot$ and $\top$, so it is clearly a subset of some Γ_j. If $i \leq 2n$, Γ_i is a four-element set. Then,

► for $1 \leq i \leq n$, $\delta(\Gamma_i, \text{in}) = \delta(\Gamma_i, \text{out}) = \Gamma_i$ (as in and out are self-loops on states of $V_{\mathcal{C}} \cup V \cup \{\bot, \top\}$);
► for $1 \leq i \leq n$,

  • $\delta(\Gamma_{n+i}, \text{in}) = \{\bot, v_i^{\mathcal{C}}, u_i, \top\} = \Gamma_i$, and
  • $\delta(\Gamma_{n+i}, \text{out}) = \{\bot, v_{i+1}^{\mathcal{C}}, u_{i+1}, \top\} = \Gamma_{i+1}$;

► for $z \in (V_{\mathcal{C}} \cup E_{\mathcal{C}}) \cup (V \cup E)$, $\delta(\Gamma_i, a_z) = \delta(\Gamma_{n+i}, a_z) = \{\bot, \top\}$, which is a subset of any Γ_j.

We have shown that sets $(\Gamma_i)_{1 \leq i \leq n+m+1}$ form a monotone decomposition of $\mathcal{D}_G$ with $n + m + 1$ sets.

We now prove the right-to-left implication. Let $(\Gamma_i)_{1 \leq i \leq n+m+1}$ be a monotone decomposition of $\mathcal{D}_G$ with $n + m + 1$ sets. Every set Γ_i contains at most two states besides $\bot$ and $\top$ (due to the chain requirement and the structure of chains in $\mathcal{D}_G$). As $V \cup E \cup \{q_{\text{init}}\}$ is an antichain with $n + m + 1$ elements, every state of this set is in exactly one set Γ_i. Due to the limited number of sets and the chain structure, states of $V_{\mathcal{C}}$ (resp. $E_{\mathcal{C}}$) need to be

in a Γ_i along with an element of V (resp. E). As V_C and V have the same cardinality, this implies that for every $i \in \{1, \dots, n\}$, there is a unique $u_i \in V$ such that v_i^C and u_i are in the same Γ_j.

We show that the sequence $(u_1, \dots, u_n)$ is a Hamiltonian cycle of G. We write $u_{n+1} = u_1$ for brevity. Let $i \in \{1, \dots, n\}$. The edge e_i^C of C_n is in some set Γ_j along with some edge $e_i = (v_i, v_{i+1}) \in E$. We have that

- ▸ $\delta(\Gamma_j, \text{in})$ contains v_i^C and v_i. As $\delta(\Gamma_j, \text{in})$ is a subset of some Γ_l, and that v_i^C is in a single set along with u_i, we deduce that $v_i = u_i$.
- ▸ similarly, from observing $\delta(\Gamma_j, \text{out})$, we deduce that $v_{i+1} = u_{i+1}$.

Therefore, $e_i = (v_i, v_{i+1}) = (u_i, u_{i+1}) \in E$. We have shown that $(u_1, \dots, u_n)$ is a Hamiltonian cycle of G, which proves the claim.

We additionally observe that objective $\mathsf{GenReach}(\mathcal{L}(\mathcal{D}_G))$ is $\mathcal{M}_{\text{triv}}$-progress-consistent. Indeed, notice that if there are $q_1, q_2 \in Q$, $w \in C^*$ such that $\delta^*(q_1, w) = q_2$ and $q_1 \prec q_2$, then $q_2 = \top$ (any progress is immediately winning). □

We now have all the ingredients to prove Theorem 7.5.1.

Theorem 7.5.1 (Complexity of MEMORY-SAFE and MEMORY-REACH) *Both MEMORY-SAFE and MEMORY-REACH are **NP**-complete.*

Proof. The MEMORY-SAFE problem is in NP (Lemma 7.6.2), and was shown to be equivalent to the MONOTONE problem, itself NP-hard (Proposition 7.6.5). This shows that MEMORY-SAFE is NP-complete.

The MEMORY-REACH problem is in NP (Corollary 7.6.4). Moreover, in Proposition 7.6.5, the finite automata considered (the $\mathcal{D}_G$ for G a directed graph) induce $\mathcal{M}_{\text{triv}}$-progress-consistent regular reachability objectives. By Theorem 7.4.8, this means that a memory structure $\mathcal{M}$ suffices for such an objective if and only if it is $\mathcal{M}$-strongly-monotone. In other words, a memory structure $\mathcal{M}$ suffices for $\mathsf{GenReach}(\mathcal{L}(\mathcal{D}_G))$ if and only if it suffices for $\mathsf{GenSafe}(\mathcal{L}(\mathcal{D}_G))$. As the problem is NP-hard for the family $\mathsf{GenSafe}(\mathcal{L}(\mathcal{D}_G))$, it is also NP-hard for the family $\mathsf{GenReach}(\mathcal{L}(\mathcal{D}_G))$. □

We remark that our proof of NP-hardness of MEMORY-REACH relies solely on the $\mathcal{M}$-strong-monotony notion. We leave as an open problem whether finding a small $\mathcal{M}$ such that a regular reachability objective is $\mathcal{M}$-progress-consistent is also NP-hard. This would be especially interesting if it held for the class of $\mathcal{M}_{\text{triv}}$-strongly-monotone objectives (i.e., for which the prefix preorder is total), as it would suggest that there is a class of automata for which finding a smallest memory structure for their induced reachability objective is harder than for their induced safety objective.

7.7 Synthesizing small memory structures in practice

7.7.1 Overview

The understanding provided by our characterizations of the memory requirements of regular reachability and safety objectives leads to a relatively straightforward way of implementing algorithms to synthesize minimal memory structures. We did so, and the result is publicly available at the link https://github.com/pvdhove/regularMemoryRequirements. We use the Python package automata-lib to represent and manipulate the automata, and the Python package PySAT [IMM18] as an interface to SAT solvers.

[IMM18]: Ignatiev et al. (2018), *PySAT: A Python Toolkit for Prototyping with SAT Oracles*

Our algorithms work as follows: given an input DFA $\mathcal{D}$ representing either a regular reachability or a regular safety objective,

▶ we perform a binary search on the number k of states of the minimal memory structure (exploiting the monotonicity of the problem: if a memory structure with k states suffices, then so does a memory structure with $k + 1$ states);

▶ for a fixed k, we encode problems MEMORY-SAFE or MEMORY-REACH into a polynomial-size propositional formula (detailed in Subsection 7.7.2 below), and check its satisfiability with a SAT solver.

For illustration purposes, we have in particular implemented the regular examples from this chapter (Examples 7.3.5, 7.4.1, and 7.4.5) — with results that of course match the theoretical analysis — and generalizations thereof (usually yielding larger automata). A user may also obtain memory requirements of a regular language by providing a regular expression, which is an easy way to try out the code.

Example 7.7.1 We illustrate one usage of the implementation on a generalization of Example 7.3.5, by requiring two extra letters e and f to be seen after seeing a, b and c, d. We display this DFA and our results in Figure 7.11 (the DFA is at the top and the memory structure at the bottom; initial states are green). We learn that a smallest memory structure for the safety objective induced by this automaton has five states (which was not obvious), and it can be displayed along with a corresponding monotone decomposition. For instance, memory state 3 induces the set $\Gamma_3 = \{1, 3, 4, 6, 7, 9\}$ in the monotone decomposition in the sense that it could be reached in parallel with automaton states $1, 3, 4, 6, 7, 9$.

Observe that there may be superfluous states in monotone decompositions: for instance, memory state 1 can be reached in parallel with automaton states 0 and 2, but not actually in parallel with 7 and 9 (even though $\Gamma_1 = \{0, 2, 7, 9\}$). This is a subtlety already hidden in the proof of equivalence between memory structures and monotone decompositions (Lemma 7.5.3): starting from a monotone decomposition, building a memory structure from it and going back to a monotone decomposition

```
D = utils.diamond_generalized(3)
```

```
M = memReq.smallest_memory_safety(D)
```

```
\Gamma_0 = ['1', '3', '4', '6', '8', '9']
\Gamma_1 = ['0', '2', '7', '9']
\Gamma_2 = ['1', '3', '5', '6', '8', '9']
\Gamma_3 = ['1', '3', '4', '6', '7', '9']
\Gamma_4 = ['1', '3', '5', '6', '7', '9']
```
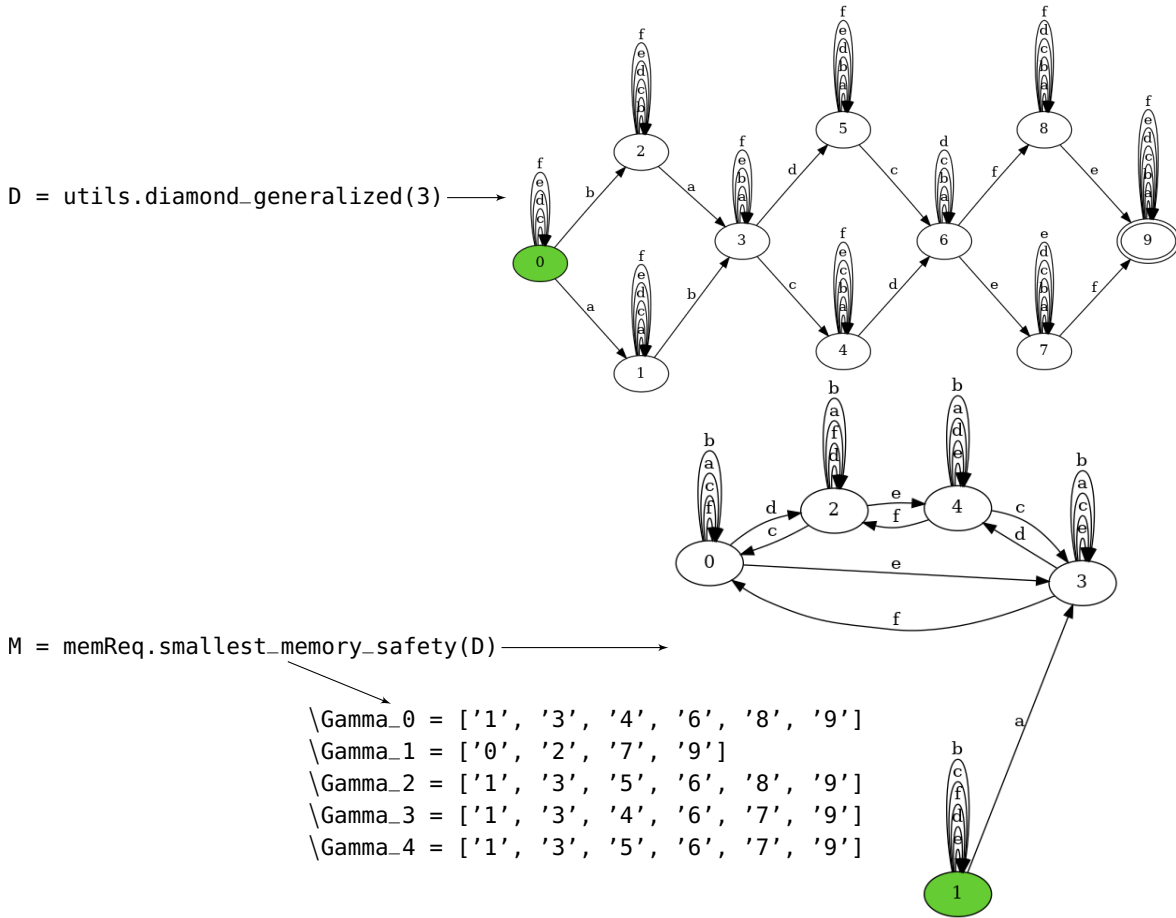
Figure 7.11: Generalization from Example 7.3.5, for which chromatic memory requirements can be computed automatically with our implementation. We give an example of Python instructions to generate the automata and the monotone decomposition.

> may not yield the exact same monotone decomposition.

7.7.2 SAT encoding

Let $\mathcal{D} = (Q, C, q_{\text{init}}, \delta, F)$ be a DFA and $k \geq 1$. We assume w.l.o.g. that $F = \{q_{\text{fin}}\}$. To encode MEMORY-SAFE as a propositional formula, we use the reformulation into *monotone decompositions* provided by Lemma 7.5.3. We consider the family of variables $(x_{q,i})_{q \in Q, 1 \leq i \leq k}$; the meaning of $x_{q,i}$ is intended to be "state q is in set Γ_i of the decomposition". The three requirements of monotone decompositions (Definition 7.5.2) can then be encoded respectively as the following propositional formulas:

(a) $\quad \bigwedge_{q \in Q} \bigvee_{1 \leq i \leq k} x_{q,i},$

(b) $\quad \bigwedge_{1 \leq i \leq k} \bigwedge_{c \in C} \bigvee_{1 \leq j \leq k} \bigwedge_{q \in Q} (x_{q,i} \implies x_{\delta(q,c),j}),$

(c) $\quad \bigwedge_{1 \leq i \leq k} \bigwedge_{q,q' \in Q^2 \text{ incomparable for } \leq} (\neg x_{q,i} \vee \neg x_{q',i}).$

From a satisfying valuation of variables $(x_{q,i})_{q,i}$, we can then recover a sufficient memory structure $\mathcal{M}$ by exploiting the proof of Lemma 7.5.3.

To encode MEMORY-REACH, we store explicitly the structure of a memory structure with k states with propositional variables. This construction is inspired from a similar encoding to minimize deterministic Büchi automata [BD14]. We fix $M = \{m_1, \ldots, m_k\}$ as the states of a possible memory structure $\mathcal{M} = (M, m_{\text{init}}, \alpha_{\text{upd}})$, and we fix arbitrarily $m_{\text{init}} = m_1$. We define a family of variables $(t_{i,c,j})_{1\leq i,j \leq k, c\in C}$ meaning that there is a transition $\alpha_{\text{upd}}(m_i, c) = m_j$ in $\mathcal{M}$. We require that $\mathcal{M}$ is complete:

$$\bigwedge_{1\leq i \leq k} \bigwedge_{c\in C} \bigvee_{1\leq j \leq k} t_{i,c,j}.$$

We do not have to enforce determinism: if a non-deterministic valuation is found, any deterministic and complete "substructure" will be a good memory structure.

We then use the NP-reformulation of $\mathcal{M}$-progress-consistency from Lemma 7.6.3; we want to know which paths are possible in the triple product involving $\mathcal{M}$ and two copies of $\mathcal{D}$. To do so, we create a family of variables $(p_{i,j,q_1,q_2,q_3,q_4})_{1\leq i,j \leq k, q_1,q_2,q_3,q_4 \in Q}$. Variable p_{i,j,q_1,q_2,q_3,q_4} indicates that $L_{m_{\text{init}},m_i} \cap L_{q_{\text{init}},q_1} \neq \emptyset$, $L_{m_{\text{init}},m_i} \cap L_{q_{\text{init}},q_3} \neq \emptyset$, and $L_{m_i,m_j} \cap L_{q_1,q_2} \cap L_{q_3,q_4} \neq \emptyset$. These constraints can be encoded with the four following formulas (explained in more detail below):

$$p_{1,1,q_{\text{init}},q_{\text{init}},q_{\text{init}},q_{\text{init}}};$$

$$\bigwedge_{1\leq i \leq k} \bigwedge_{q\in Q} \bigwedge_{c\in C} \bigwedge_{1\leq j \leq k} (p_{i,i,q,q,q,q} \wedge t_{i,c,j} \implies p_{j,j,\delta(q,c),\delta(q,c),\delta(q,c),\delta(q,c)});$$

$$\bigwedge_{1\leq i \leq k} \bigwedge_{q,q'\in Q} (p_{i,i,q,q,q,q} \wedge p_{i,i,q',q',q',q'} \implies p_{i,i,q,q,q',q'});$$

$$\bigwedge_{1\leq i,j,l \leq k} \bigwedge_{q_1,q_2,q_3,q_4 \in Q} \bigwedge_{c\in C} (p_{i,j,q_1,q_2,q_3,q_4} \wedge t_{j,c,l}$$
$$\implies p_{i,l,q_1,\delta(q_2,c),q_3,\delta(q_4,c)}).$$

The first formula simply defines the initial state in the product. The second one computes the reachable states of the product between $\mathcal{M}$ and $\mathcal{D}$, i.e., couples (i,q) such that $L_{m_{\text{init}},m_i} \cap L_{q_{\text{init}},q} \neq \emptyset$ (this would be doable with variables with only two indices i and q). The third one extends it to all triplets (i,q,q') such that $L_{m_{\text{init}},m_i} \cap L_{q_{\text{init}},q} \neq \emptyset$ and $L_{m_{\text{init}},m_i} \cap L_{q_{\text{init}},q'} \neq \emptyset$. The fourth one augments the paths in the triple product: if there is already a word w from m_i to m_j, from q_1 to q_2, and from q_3 to q_4, then for all colors c, if $\alpha_{\text{upd}}(m_j, c) = m_l$, word wc goes from m_i to m_l, from q_1 to $\delta(q_2, c)$, and from q_3 to $\alpha_{\text{upd}}(q_4, c)$.

With these variables, we can then encode the constraint of Lemma 7.6.3, which is equivalent to $\mathcal{M}$-progress-consistency:

$$\bigwedge_{1\leq i \leq k} \bigwedge_{\substack{q_1,q_2\in Q \\ q_2 \neq q_{\text{fin}}, q_1 < q_2}} (p_{1,i,q_{\text{init}},q_1,q_{\text{init}},q_1} \implies \neg p_{i,i,q_1,q_2,q_2,q_2}).$$

This only deals with $\mathcal{M}$-progress-consistency; to enforce $\mathcal{M}$-strong-monotony at the same time, notice that $p_{i,i,q,q,q}$ has the same meaning as $x_{q,i}$ above. We can simply add the three constraints already discussed for regular safety objectives to our formula. As variables $t_{i,c,j}$ encode the transitions of a memory structure, it is straightforward to recover a memory structure from a satisfying valuation.

These formulas are not all in *conjunctive normal form*, which is usually required by SAT solvers; they can be converted into such formulas using the Tseitin transform [Tse70], which increases the number of variables and clauses polynomially — details about this transformation are straightforward and are documented in the source code. These formulas in conjunctive normal form contain in the end, if n is the number of states of the input DFA:

- ▶ for Memory-Safe, $kn + k^2 \cdot (n+1) \cdot |C|$ variables and $\mathcal{O}(kn^2 + k^2 n |C|)$ clauses;
- ▶ for Memory-Reach, $k^2 n^4 + k^2 \cdot (n+2) \cdot |C|$ variables and $\mathcal{O}(k^3 n^4 |C|)$ clauses.

Most clauses have size 2 or 3, and a small amount of clauses have size k in both cases.

7.8 Wrap-up

We have characterized the minimal memory structures sufficient to play optimally for regular reachability and safety objectives, a simple class of ω-regular objectives. Some of our results apply more generally to general reachability and safety objectives (usually called *topologically open* and *topologically closed* objectives), which are the objectives at the first level of the Borel hierarchy. Throughout our characterizations, we were able to prove that decision problems related to finding minimal memory structures for regular objectives are NP-complete. Our characterizations were encoded into a SAT solver that automatically generates minimal memory structures for both players given a DFA as an input.

This chapter can be seen as one step toward understanding more generally the memory requirements of all ω-regular objectives, as well as synthesizing minimal memory structures for them. The chaotic memory requirements of regular reachability objectives are still unknown (as opposed to the ones of regular safety objectives [CFH14; CFH22]), as well as the chromatic memory requirements of larger classes of ω-regular objectives.

In the next chapter, we move on to a more general class of ω-regular objectives, at the expense of only giving a complete understanding of half-positionality, and not of full memory requirements.

Half-positional objectives recognized by deterministic Büchi automata

<div style="text-align: right;">

8

</div>

In this chapter, we focus on *half-positionality*, i.e., the property of an objective for which the first player does not need memory to implement winning strategies. Even though there are characterizations of objectives that admit memoryless optimal strategies for *both* players (see Chapter 3), focusing on just one player appears to be a more difficult question. Chapter 7 gave us, as a special case, a complete understanding of half-positionality for *regular* objectives, a simple class of ω-regular objectives. In order to get closer to the whole class of ω-regular objectives, we consider a class of objectives one step further.

We characterize objectives recognizable by *deterministic Büchi automata* (a class of ω-regular objectives strictly encompassing the regular objectives from the previous chapter) that are half-positional, both over finite and infinite graphs. Our characterization consists of three natural conditions linked to the language-theoretic *right congruence*. Furthermore, this characterization yields a polynomial-time algorithm to decide half-positionality of an objective recognized by a given deterministic Büchi automaton.

Compared to Chapter 7, we therefore deal with a larger class of objectives, but give a complete understanding of a more restricted class of strategies. To understand half-positionality, we will revisit in particular the necessary $\mathcal{M}_{\text{triv}}$-strong-monotony and the $\mathcal{M}_{\text{triv}}$-progress-consistency properties introduced in the previous chapter, but we will also need additional properties.

[BCRV22]: Bouyer et al. (2022), *Half-Positional Objectives Recognized by Deterministic Büchi Automata*

8.1 Introduction

Memoryless determinacy and half-positionality.   At multiple occasions in this thesis, we have given ways to understand the memory requirements of objectives for *both* players simultaneously. If we look at arguably the simplest kind of strategies — memoryless ones — we have on the one hand *memoryless determinacy* (the property of objectives admitting memoryless optimal strategies for *both* players) and on the other hand *half-positionality* (the same for just *one* player). We have discussed in Chapter 3 characterizations allowing us to understand memoryless determinacy of objectives on both finite [GZ05] and infinite [CN06] graphs.

Yet, there exist many objectives and combinations thereof for which one player, but not both, has memoryless optimal strategies (e.g., Rabin conditions [KK91; Kla94], mean-payoff parity [CHJ05], energy parity [CD12], some window objectives [CDRR15; BHR16], energy mean-payoff [BHRR19]), and to which these results do not apply.

Various attempts have been made to understand common underlying properties of half-positional objectives and to provide sufficient conditions [Kop06; Kop07; Kop08; BFMM11], but little more was known until the recent work of Ohlmann [Ohl23] (discussed below). These sufficient conditions are not general enough to prove half-positionality of some very simple objectives, even ω-regular ones [BFMM11, Lemma 13] — this is what we intend to tackle here. Furthermore, multiple questions concerning half-positionality remain open. For instance, in [Kop08], Kopczyński conjectured that *prefix-independent* half-positional objectives are closed under finite union (this conjecture was recently refuted for games on finite graphs [Koz22a], but is still unsolved for games on infinite graphs). Also, Kopczyński showed that given a deterministic parity automaton recognizing a prefix-independent objective W, we can decide if W is half-positional over finite arenas [Kop07]. However, the time complexity of his algorithm is $\mathcal{O}(n^{\mathcal{O}(n^2)})$, where n is the number of states of the automaton. This complexity comes from a reduction of half-positionality over all finite arenas to half-positionality over a finite (but large) number of finite arenas, which can be enumerated. It is unknown whether half-positionality can be decided in polynomial time, and no algorithm is known for the non-prefix-independent case.

Deterministic Büchi automata.   In this chapter, we focus on the proper subclass of ω-regular objectives recognized by *deterministic Büchi automata* (DBAs), that we call *DBA-recognizable*. DBA-recognizable objectives correspond to the ω-regular objectives that can be written as a countable intersection of open objectives (that is, which are the G_δ sets of the Borel hierarchy) or, equivalently, that are the limit language of a regular language of finite words [PP04]. Deciding the winner of a game with a DBA-recognizable objective is doable in polynomial time in the size of the

[GZ05]: Gimbert et al. (2005), *Games Where You Can Play Optimally Without Any Memory*

[CN06]: Colcombet et al. (2006), *On the positional determinacy of edge-labeled games*

[KK91]: Klarlund et al. (1991), *Rabin Measures and Their Applications to Fairness and Automata Theory*

[Kla94]: Klarlund (1994), *Progress Measures, Immediate Determinacy, and a Subset Construction for Tree Automata*

[CHJ05]: Chatterjee et al. (2005), *Mean-Payoff Parity Games*

[CD12]: Chatterjee et al. (2012), *Energy parity games*

[CDRR15]: Chatterjee et al. (2015), *Looking at mean-payoff and total-payoff through windows*

[BHR16]: Bruyère et al. (2016), *Window parity games: an alternative approach toward parity games with time bounds*

[BHRR19]: Bruyère et al. (2019), *Energy Mean-Payoff Games*

[Kop06]: Kopczyński (2006), *Half-Positional Determinacy of Infinite Games*

[Kop07]: Kopczyński (2007), *Omega-Regular Half-Positional Winning Conditions*

[Kop08]: Kopczyński (2008), *Half-positional Determinacy of Infinite Games*

[BFMM11]: Bianco et al. (2011), *Exploring the boundary of half-positionality*

[Ohl23]: Ohlmann (2023), *Characterizing Positionality in Games of Infinite Duration over Infinite Graphs*

[Koz22a]: Kozachinskiy (2022), *Energy Games over Totally Ordered Groups*

[PP04]: Perrin et al. (2004), *Infinite words – automata, semigroups, logic and games*

arena and the DBA (by solving a Büchi game on the product of the arena and the DBA [BCJ18]).

We will make use of two central technical tools: the first is the notion of *right congruence*, already discussed at length and used heavily in Chapter 5; the second is the notion of *universal graph*, a novel tool to solve games on graphs.

[BCJ18]: Bloem et al. (2018), *Graph Games and Reactive Synthesis*
See Definition 2.8.3 for a definition of the right congruence.

Universal graphs. A recent advance in the study of half-positionality is the introduction of *well-monotonic universal graphs*, combinatorial structures that can be used to provide a witness of winning strategies in games with a half-positional objective. Recently, Ohlmann [Ohl23] has shown that the existence of a *well-monotonic universal graph* for an objective W exactly characterizes half-positionality (under minor technical assumptions on W). Moreover, under these assumptions, a wide class of algorithms, called *value iteration algorithms*, can be applied to solve any game with a half-positional objective [Ohl21; CFGO22].

[Ohl23]: Ohlmann (2023), *Characterizing Positionality in Games of Infinite Duration over Infinite Graphs*

Although it brings insight into the structure of half-positional objectives, showing half-positionality through the use of universal graphs is not always straightforward and has not yet been applied in a systematic way to ω-regular objectives.

[Ohl21]: Ohlmann (2021), *Monotonic graphs for parity and mean-payoff games*
[CFGO22]: Colcombet et al. (2022), *The Theory of Universal Graphs for Infinite Duration Games*

Contributions. Our main contribution is a *characterization* of the DBA-recognizable objectives that are half-positional, through a conjunction of three easy-to-check conditions (Theorem 8.3.7). The first two conditions are specializations of the $\mathcal{M}$-strong-monotony and $\mathcal{M}$-progress-consistency already introduced in Chapter 7; these more general properties could deal with arbitrary memory structures, whereas here we only need the "memoryless memory structure" $\mathcal{M}_{\text{triv}}$. The three conditions can be stated as follows:

(a) *($\mathcal{M}_{\text{triv}}$-strong-monotony)* The equivalence classes for the right congruence are *totally* ordered w.r.t. inclusion of their winning continuations.
(b) *($\mathcal{M}_{\text{triv}}$-progress-consistency)* Whenever the set of winning continuations of a finite word w_1 is a proper subset of the set of winning continuations of a concatenation $w_1 w_2$, the word $w_1 (w_2)^{\omega}$ produced by repeating infinitely often w_2 is winning.
(c) The objective has to be recognizable by a DBA using the structure of its prefix classifier.

A few examples of simple DBA-recognizable objectives that were not encompassed by previous half-positionality criteria [Kop06; BFMM11] are, e.g., *weak parity* objectives [Tho08], and "reaching a color twice in a row or infinitely often" (Example 8.3.3), which is half-positional but not memoryless-determined, and whose half-positionality is straightforward using our characterization.

[Kop06]: Kopczyński (2006), *Half-Positional Determinacy of Infinite Games*
[BFMM11]: Bianco et al. (2011), *Exploring the boundary of half-positionality*
[Tho08]: Thomas (2008), *Church's Problem and a Tour through Automata Theory*

Various corollaries with practical and theoretical interest follow from our characterization.

▶ We obtain a painless path to show (by checking each of the three conditions) that given a deterministic Büchi automaton, the half-positionality of the objective it recognizes is decidable in time $\mathcal{O}(k^2 \cdot n^4)$, where k is the number of colors and n is the number of states of the DBA (Subsection 8.3.3).

▶ Prefix-independent DBA-recognizable half-positional objectives are exactly the very simple Büchi conditions Büchi(C') for some $C' \subseteq C$, which consist of all the infinite words seeing infinitely many times colors in C' (Proposition 8.3.8). In particular, Kopczyński's conjecture about finite unions of half-positional objectives trivializes for DBA-recognizable objectives (the finite union of Büchi conditions is a Büchi condition).

▶ We obtain a *finite-to-infinite* and a strong *one-to-two-player lift* (Theorem 8.3.11): in order to check that a DBA-recognizable objective is half-positional over arbitrary — possibly two-player and infinite — game graphs, it suffices to check the existence of memoryless optimal strategies over *finite one-player* graphs where all the vertices are controlled by $\mathcal{P}_1$.

Technical overview. The necessity of Conditions (a) and (b) for half-positionality over one-player arenas (for ω-regular objectives, even over *finite* one-player arenas) was already shown in Propositions 7.3.1 and 7.4.4. Condition (c) has been studied multiple times in the language-theoretic literature, both for itself and for minimization and learning algorithms [Sta83; LeS90; MS97; AFS20; BL21; AF21]; we also gave a strategic characterization of it for deterministic *parity* automata in Corollary 5.4.6. As an example, all deterministic *weak* automata (a restriction on DBAs) satisfy Condition (c) [Sta83; AF21]. We prove that Condition (c) is necessary for half-positionality of DBA-recognizable objectives, but not for all (even ω-regular) objectives in general (see Example 8.3.6). The proof of its necessity is more involved than for the first two conditions, and will build on automata-theoretic ideas introduced for *good-for-games coBüchi automata* [AK19; AK20; AK22].

Together, the three conditions are sufficient for half-positionality of DBA-recognizable objectives: the proof of sufficiency uses the theory of universal graphs, and consists in building a family of well-monotonic universal graphs [Ohl23] for objectives satisfying the three properties.

Chapter structure. We first introduce technical notions from [AK22] to manipulate DBAs in Section 8.2. Our main contributions are presented in Section 8.3: we introduce and discuss the three conditions used in our results, then we state our main characterization (Theorem 8.3.7) and some corollaries, and we end with an explanation on how to use

[Sta83]: Staiger (1983), *Finite-State ω-Languages*

[LeS90]: Le Saëc (1990), *Saturating right congruences*

[MS97]: Maler et al. (1997), *On Syntactic Congruences for Omega-Languages*

[AFS20]: Angluin et al. (2020), *Polynomial Identification of ω-Automata*

[BL21]: Bohn et al. (2021), *Constructing Deterministic ω-Automata from Examples by an Extension of the RPNI Algorithm*

[AF21]: Angluin et al. (2021), *Regular ω-languages with an informative right congruence*

[AK19]: Abu Radi et al. (2019), *Minimizing GFG Transition-Based Automata*

[AK20]: Abu Radi et al. (2020), *Canonicity in GFG and Transition-Based Automata*

[AK22]: Abu Radi et al. (2022), *Minimization and Canonization of GFG Transition-Based Automata*

[Ohl23]: Ohlmann (2023), *Characterizing Positionality in Games of Infinite Duration over Infinite Graphs*

the characterization to decide half-positionality of DBA-recognizable objectives in polynomial time. Sections 8.4 and 8.5 contain the proof of Theorem 8.3.7: the former shows the necessity of Condition (c) for half-positionality of DBA-recognizable objectives, and the latter shows their sufficiency through the use of universal graphs.

Notions. We do not devote a section to additional preliminaries in this chapter, but we will especially make use of the following terminology defined in Chapter 2: *automaton structures* (Definition 2.7.1), *deterministic Büchi automata* (DBAs — Definition 2.7.3), *prefix preorder* (Definition 2.8.2), *right congruence* (Definition 2.8.3), and *prefix classifier* (Definition 2.8.11). We also extend the notation of the objective $\mathsf{Büchi}(a)$ (containing words that see color $a \in C$ infinitely often) to $\mathsf{Büchi}(C')$ for $C' \subseteq C$ (containing words that, infinitely often, see a color that belongs to C'). Objectives that can be recognized by a DBA are called *DBA-recognizable*.

We recall that an objective W is *half-positional (over resp. finite, countable, finitely branching, one-player arenas)* if for all (resp. finite, countable, finitely branching, one-player) arenas $\mathcal{A}$, $\mathcal{P}_1$ has a memoryless optimal strategy (i.e., an optimal strategy based on $\mathcal{M}_{\mathsf{triv}}$) in game $(\mathcal{A}, W)$.

For a DBA $\mathcal{B} = (Q, C, q_{\mathsf{init}}, \delta, B)$, a state $q \in Q$, and an infinite word $w = c_1 c_2 \ldots \in C^\omega$, we write $\mathcal{B}(q, w) \in (Q \times C)^\omega$ for the *infinite run of $\mathcal{B}$ on w from q*, i.e., the sequence $(q_0, c_1)(q_1, c_2) \ldots$ with $q_0 = q$ and $q_i = \delta(q_{i-1}, c_i)$ for $i \geq 1$. If $w \in C^*$, we define the *finite run of $\mathcal{B}$ on w from q* in a similar way.

8.2 Saturating Büchi automata

We describe a "normal form" of deterministic Büchi automata that has good properties and that will be used throughout multiple sections of this chapter.

Let $\mathcal{B} = (Q, C, q_{\mathsf{init}}, \delta, B)$ be a DBA. We say that a finite run of $\mathcal{B}$ is *B-free* if it does not contain any transition from B. For $q \in Q$, we define

$$B\text{-}\mathsf{Free}_\mathcal{B}(q) = \{w \in C^* \mid \mathcal{B}(q, w) \text{ is } B\text{-free}\},$$
$$B\text{-}\mathsf{FreeCycles}_\mathcal{B}(q) = \{w \in C^* \mid w \in B\text{-}\mathsf{Free}_\mathcal{B}(q) \text{ and } \delta^*(q, w) = q\}.$$

We call the words in the first set the *B-free words from q*, and the words in the second set the *B-free cycles from q*. We state an important property of B-free words; *having the same B-free words* is a kind of congruence on the states of the automata (if we restrict words read to B-free words).

> **Lemma 8.2.1** *Let $q_1, q_2 \in Q$ be such that $B\text{-}\mathsf{Free}_\mathcal{B}(q_1) = B\text{-}\mathsf{Free}_\mathcal{B}(q_2)$. Then for all $w \in B\text{-}\mathsf{Free}_\mathcal{B}(q_1)$, $B\text{-}\mathsf{Free}_\mathcal{B}(\delta^*(q_1, w)) = B\text{-}\mathsf{Free}_\mathcal{B}(\delta^*(q_2, w))$.*
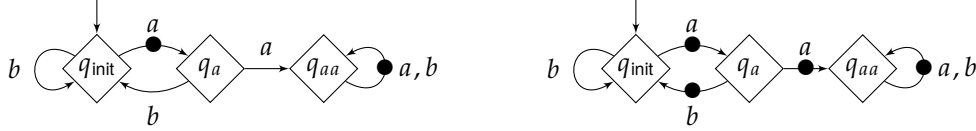
Figure 8.1: DBA (left) and its unique saturation (right). We recall that transitions labeled with a • symbol are the Büchi transitions.

Proof. Let $w \in B\text{-Free}_{\mathcal{B}}(q_1)$. Let $q'_1 = \delta^*(q_1, w)$, $q'_2 = \delta^*(q_2, w)$, and $w' \in B\text{-Free}_{\mathcal{B}}(q'_1)$. Since both runs $\mathcal{B}(q_1, w)$ and $\mathcal{B}(q'_1, w')$ are B-free, we have $ww' \in B\text{-Free}_{\mathcal{B}}(q_1) = B\text{-Free}_{\mathcal{B}}(q_2)$. Therefore, the run $\mathcal{B}(q_2, ww')$ is B-free, so the run $\mathcal{B}(q'_2, w')$ is B-free as well and $w' \in B\text{-Free}_{\mathcal{B}}(q'_2)$. $\square$

In what follows, the "normal form" of Büchi automata that we will define satisfies that any B-free path can be extended to a B-free cycle. Such a normal form can be produced by *saturating* a given DBA $\mathcal{B}$ with Büchi transitions [KS15; AK19; AK20]. To do so, we add to B all transitions that do not appear in a B-free cycle of $\mathcal{B}$. Cycles that are B-free can be easily identified by decomposing in strongly connected components the structure obtained by removing the Büchi transitions from $\mathcal{B}$.

[KS15]: Kuperberg et al. (2015), *On Determinisation of Good-for-Games Automata*
[AK19]: Abu Radi et al. (2019), *Minimizing GFG Transition-Based Automata*
[AK20]: Abu Radi et al. (2020), *Canonicity in GFG and Transition-Based Automata*

We say that $\mathcal{B} = (Q, C, q_{\text{init}}, \delta, B)$ is *saturated* if for every $B' \supsetneq B$, the automaton obtained by replacing B with B' does not recognize $\mathcal{L}(\mathcal{B})$.

A *B-free component* of $\mathcal{B}$ is a strongly connected component of the graph obtained by removing the Büchi transitions from $\mathcal{B}$. Formally, let $\Delta_{\text{noB}} = \{(q, c, \delta(q, c)) \in Q \times C \times Q \mid (q, c) \notin B\}$; a B-free component is a strongly connected component of the graph (Q, Δ_{noB}).

> **Example 8.2.2** In Figure 8.1 (left), we show a DBA with a single B-free component consisting of state q_{init} and transition (q_{init}, b). It turns out that we can make any transition not in a B-free component into a Büchi transition without changing the language recognized by the automaton (as illustrated with the *saturated* automaton in Figure 8.1, right). Let us consider transition (q_a, b): the intuition is that it is not possible to see this transition infinitely often without seeing the Büchi transition (q_{init}, a) infinitely often. Hence, we might as well make (q_a, b) a Büchi transition, and it does not change which runs are accepted.

We generalize the reasoning used in this example.

> **Lemma 8.2.3** *Let $\mathcal{B} = (Q, C, q_{\text{init}}, \delta, B)$ be a DBA. There is a unique set $B_{\text{sat}} \subseteq Q \times C$ such that the DBA $\mathcal{B}_{\text{sat}} = (Q, C, q_{\text{init}}, \delta, B_{\text{sat}})$ satisfies that:*
>
> 1. *$\mathcal{L}(\mathcal{B}_{\text{sat}}) = \mathcal{L}(\mathcal{B})$;*
> 2. *$\mathcal{B}_{\text{sat}}$ is saturated.*
>
> *Moreover, B_{sat} is the set of transitions not appearing in any B-free component of $\mathcal{B}$, and it can be computed in time $\mathcal{O}(|C| \cdot |Q|)$ when C is finite.*

Proof. We first prove the existence of such a B_{sat}. Let $\Delta_{\text{freeComp}} \subseteq Q \times C \times Q$ be the set of transitions appearing in a B-free component of $\mathcal{B}$ (it is a subset of Δ_{noB}). We consider the automaton $\mathcal{B}_{\text{sat}}$ whose Büchi transitions are those that do not appear in any B-free component, that is, we let

$$B_{\text{sat}} = (Q \times C) \setminus \{(q,c) \mid (q,c,\delta(q,c)) \in \Delta_{\text{freeComp}}\} \text{ and}$$
$$\mathcal{B}_{\text{sat}} = (Q, C, q_{\text{init}}, \delta, B_{\text{sat}}).$$

We show that $\mathcal{L}(\mathcal{B}_{\text{sat}}) = \mathcal{L}(\mathcal{B})$. Since $B \subseteq B_{\text{sat}}$, it holds that $\mathcal{L}(\mathcal{B}) \subseteq \mathcal{L}(\mathcal{B}_{\text{sat}})$. For the other inclusion, let $w \notin \mathcal{L}(\mathcal{B})$. There are $w_0 \in C^*, w' \in C^\omega$ such that $w = w_0 w'$ and the infinite run $\mathcal{B}(q_0, w')$ produced by reading w' from $q_0 = \delta^*(q_{\text{init}}, w_0)$ does not visit any Büchi transition. In particular, the infinite run $\mathcal{B}(q_0, w')$ is an infinite path in the finite graph $(Q, \Delta_{\text{safe}})$. This implies that eventually, run $\mathcal{B}(q_0, w')$ reaches and stays in the same strongly connected component of graph $(Q, \Delta_{\text{safe}})$. Formally, there are $w_1 \in C^*$ and $w_2 \in C^\omega$ such that $w' = w_1 w_2$ and the infinite run $\mathcal{B}(q_1, w_2)$ produced by reading w_2 from $q_1 = \delta^*(q_0, w_1)$ lies entirely in some B-free component. Thus, all transitions in $\mathcal{B}(q_1, w_2)$ are in Δ_{freeComp} and do not induce transitions in B_{sat}. Therefore, $\mathcal{B}_{\text{sat}}(q_{\text{init}}, w)$ is also a rejecting run, and $w \notin \mathcal{L}(\mathcal{B}_{\text{sat}})$.

We prove that $\mathcal{B}_{\text{sat}}$ is saturated and the uniqueness of B_{sat} at the same time. Let B' be another set of transitions such that $B' \not\subseteq B_{\text{sat}}$ and let $\mathcal{B}'$ be the automaton obtained by replacing B_{sat} with B' in $\mathcal{B}_{\text{sat}}$. Let $(q,c) \in B' \setminus B_{\text{sat}}$. Since $(q,c) \notin B_{\text{sat}}$, transition (q,c) is in a B-free component. We can therefore consider a word $w \in B\text{-FreeCycles}_{\mathcal{B}_{\text{sat}}}(q)$ that is a B-free cycle from q and that uses the transition (q,c). Let $w_0 \in C^*$ such that $\delta^*(q_{\text{init}}, w_0) = q$. Then, $w_0 w^\omega \notin \mathcal{L}(\mathcal{B}_{\text{sat}})$, whereas $w_0 w^\omega \in \mathcal{L}(\mathcal{B}')$, so $\mathcal{B}'$ does not recognize the same objective as $\mathcal{B}_{\text{sat}}$.

When C is finite, the set of transitions $\mathcal{B}_{\text{sat}}$ can be computed in time $\mathcal{O}(|C| \cdot |Q|)$, as it consists of decomposing a graph with Q vertices and at most $|C| \cdot |Q|$ edges into strongly connected components [Tar72]. □

[Tar72]: Tarjan (1972), *Depth-First Search and Linear Graph Algorithms*

The following simple lemma follows, which holds in saturated DBAs: every word that is a B-free *word* from a state can be completed into a B-free *cycle* from the same state. This is a key technical lemma used many times in the upcoming proofs.

> **Lemma 8.2.4** *Let* $\mathcal{B} = (Q, C, q_{\text{init}}, \delta, B)$ *be a* saturated *DBA. Let* $q \in Q$ *and* $w \in B\text{-Free}_{\mathcal{B}}(q)$. *There exists* $w' \in C^*$ *such that* $ww' \in B\text{-FreeCycles}_{\mathcal{B}}(q)$.

Proof. Let $q' = \delta^*(q, w)$. Thanks to the saturation property and by Lemma 8.2.3, B contains all transitions that do not belong to a B-free component. This implies that any two states connected by a B-free run are in the same B-free component. In particular, as q' is reachable from q through a B-free run, q' must belong to the same B-free component as q.

Therefore, there exists a B-free run from q' to q. Taking the word $w' \in C^*$ labeling this run, we obtain the desired result. $\square$

8.3 Half-positionality of DBA-recognizable objectives

In this section, we present our main contribution in Theorem 8.3.7, by giving three conditions that exactly characterize half-positional DBA-recognizable objectives. These conditions are presented in Subsection 8.3.1. Theorem 8.3.7 and several consequences of it are stated in Subsection 8.3.2 (the proof of Theorem 8.3.7 is postponed to Sections 8.4 and 8.5). In Subsection 8.3.3, we use this characterization to show that we can decide the half-positionality of a DBA in polynomial time.

8.3.1 Three conditions for half-positionality

We define the three conditions of objectives at the core of our characterization.

> **Condition 1** (Total prefix preorder) *We say that an objective $W \subseteq C^\omega$ has a total prefix preorder if for all $w_1, w_2 \in C^*$, $w_1 \le w_2$ or $w_2 \le w_1$.*

For an objective W, *having a total prefix preorder* is equivalent to being $\mathcal{M}_{\text{triv}}$-strongly-monotone (Section 7.3): all finite words must be comparable for the prefix preorder $\le$ of W, which, for $\mathcal{M}_{\text{triv}} = (\{m_{\text{init}}\}, m_{\text{init}}, \alpha_{\text{upd}})$, is equivalent to requiring that all words in $L_{m_{\text{init}}, m_{\text{init}}} = C^*$ are comparable for $\le$. This implies (Lemma 4.3.9) that having a total prefix preorder is "symmetric": an objective has a total prefix preorder if and only if its complement has a total prefix preorder too.

> **Example 8.3.1** (Non-total prefix preorder) We have already seen multiple objectives that do not have a total prefix preorder (for instance, objective $W = \text{Reach}(a) \cap \text{Reach}(b)$ studied in Subsection 4.4.2). We give another example of a DBA-recognizable objective that will satisfy the other two properties of the characterization.
>
> Let $C = \{a, b\}$. We consider the objective W recognized by the DBA $\mathcal{B}$ depicted in Figure 8.2 (left). It consists of the infinite words starting with aa or bb. This objective does not have a total prefix preorder: words a and b are incomparable for $\le$. Indeed, a^ω is winning after a but not after b, and b^ω is winning after b but not after a. In terms of automaton states, we have that q_a and q_b are incomparable for $\le_\mathcal{B}$. This objective is not half-positional, as witnessed by the arena on the right of Figure 8.2. In this arena, $\mathcal{P}_1$ is able to win when the game starts in v_1 by playing a in v_3, and when the game starts in v_2 by playing b. However, no memoryless strategy wins from both v_1 and v_2.
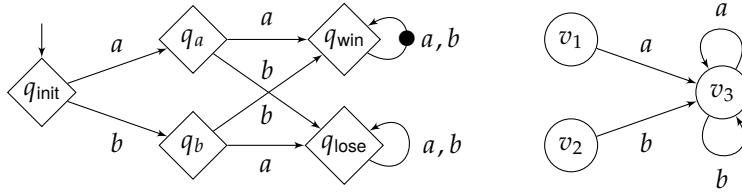
A straightforward result for an objective W recognized by a DBA $\mathcal{B}$ is that it has a total prefix preorder if and only if the (reachable) states of $\mathcal{B}$ are totally ordered for $\preceq_\mathcal{B}$.

The second condition is the specialization of $\mathcal{M}$-progress-consistency (Section 7.4) to the case $\mathcal{M} = \mathcal{M}_{\text{triv}}$. We rewrite it explicitly here for clarity.

Condition 2 ($\mathcal{M}_{\text{triv}}$-progress-consistency) *An objective W is $\mathcal{M}_{\text{triv}}$-progress-consistent if and only if for all $w_1 \in C^*$ and $w_2 \in C^+$ such that $w_1 \prec w_1 w_2$, we have $w_1(w_2)^\omega \in W$.*

Intuitively, this means that whenever a word w_2 can be used to make progress after seeing a word w_1 (in the sense of getting to a position in which more continuations are winning), then repeating this word has to be winning.

Example 8.3.2 (Non-$\mathcal{M}_{\text{triv}}$-progress-consistent objective) Let $C = \{a, b\}$. We consider the objective $W = C^* aa C^\omega$ recognized by the DBA with three states in Figure 8.3 (left). This objective contains the words seeing, at some point, twice the color a in a row. Notice that the prefix preorder of this objective is total ($q_{\text{init}} \prec q_a \prec q_{aa}$). This objective is not $\mathcal{M}_{\text{triv}}$-progress-consistent: we have $\varepsilon \prec ba$, but $(ba)^\omega \notin W$. This objective is not half-positional: if $\mathcal{P}_1$ plays in an arena with a choice among two cycles ba and ab depicted in Figure 8.3 (right), it is possible to win by playing ba and then ab, but a memoryless strategy can only achieve words $(ba)^\omega$ or $(ab)^\omega$, which are both losing.

Example 8.3.3 ($\mathcal{M}_{\text{triv}}$-progress-consistent objective) We consider a slight modification to the previous example by adding two Büchi transitions: see the DBA in Figure 8.4. The objective recognized by this DBA is $W = \text{Büchi}(a) \cup C^* aa C^\omega$: W contains the words seeing a infinitely often, or that see a twice in a row at some point. Properties of this objective were already sketched in Example 6.1.1.
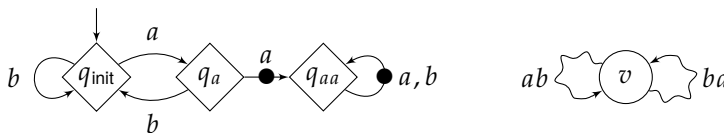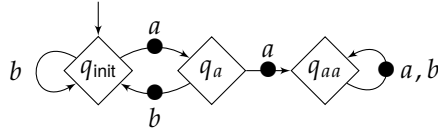
The equivalence classes for $\sim_W$ are $q_{\text{init}}^{-1}W = W$, $q_a^{-1}W = aC^\omega \cup W$ and $q_{aa}^{-1}W = C^\omega$. This objective is $\mathcal{M}_{\text{triv}}$-progress-consistent: any word reaching q_{aa} is straightforwardly accepted when repeated infinitely often, and any word w such that $\delta^*(q_{\text{init}}, w) = q_a$ necessarily contains at least one a, and thus is accepted when repeated infinitely often. Objective W is half-positional, which will be readily shown with our upcoming characterization (Theorem 8.3.7).

Here, notice that the complement $\overline{W}$ of W is not $\mathcal{M}_{\text{triv}}$-progress-consistent. Indeed, $a \prec_{\overline{W}} a(bab)$, but $a(bab)^\omega \notin \overline{W}$. Unlike having a total prefix preorder, $\mathcal{M}_{\text{triv}}$-progress-consistency can hold for an objective but not its complement.

Note that half-positionality of W cannot be shown using existing half-positionality criteria [Kop06; BFMM11] (it is neither prefix-independent nor *concave*) or memoryless-determinacy criteria, as it is simply not memoryless-determined.

[Kop06]: Kopczyński (2006), *Half-Positional Determinacy of Infinite Games*
[BFMM11]: Bianco et al. (2011), *Exploring the boundary of half-positionality*

The third condition is the novelty of this chapter.

Condition 3 (Recognizability by the prefix classifier) *Being* recognized by a Büchi automaton built on top of the prefix classifier *is our third condition. In other words, for a DBA-recognizable objective $W \subseteq C^\omega$ and its prefix classifier $\mathcal{S}_W = (Q_W, C, q_{\text{init}}^W, \delta_W)$, this condition requires that there exists $B_\sim \subseteq Q_W \times C$ such that W is recognized by DBA $(Q_W, C, q_{\text{init}}^W, \delta_W, B_\sim)$.*

We show an example of a DBA-recognizable objective satisfying the first two conditions (total prefix preorder and $\mathcal{M}_{\text{triv}}$-progress-consistency), but not this third condition, and which is not half-positional.

Example 8.3.4 (Not recognizable by the prefix classifier) Let $C = \{a, b\}$. We consider the objective $W = \text{Büchi}(a) \cap \text{Büchi}(b)$ recognized by the DBA in Figure 8.5. This objective is prefix-independent: as such, there is only one equivalence class for $\sim$. This implies that the prefix preorder is total, and that W is $\mathcal{M}_{\text{triv}}$-progress-consistent (the premise of the $\mathcal{M}_{\text{triv}}$-progress-consistency property can never be true). This objective is not half-positional, as witnessed by the arena in Figure 8.5 (right): $\mathcal{P}_1$ has a winning strategy from v, but it needs to take infinitely often both a and b.

Any DBA recognizing this objective has at least two states, but all their (reachable) states are equivalent for $\sim$: no matter the state we choose as an initial state, the recognized objective is the same. As it is prefix-independent, its prefix classifier $\mathcal{S}_W$ has only one state.
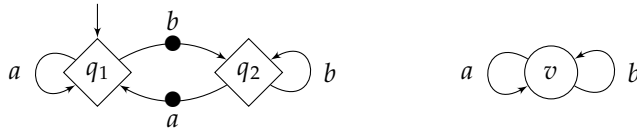
As will be shown formally, being recognized by a DBA built on top of the prefix classifier is necessary for half-positionality of *DBA-recognizable* objectives over finite one-player arenas.

> **Remark 8.3.5** All *regular objectives* (Chapter 7) satisfy the property of being recognizable by their prefix classifier, which is due to Myhill-Nerode theorem [Ner58]. This may give some insight as to why this condition was not needed in Chapter 7.

[Ner58]: Nerode (1958), *Linear Automaton Transformations*

Unlike the two other conditions, this third condition is in general not necessary for half-positionality of general objectives, including objectives recognized by other standard classes of automata on infinite words.

> **Example 8.3.6** We consider the complement $\overline{W}$ of the objective $W = $ Büchi(a)∩Büchi(b) of Example 8.3.4, which consists of the words ending with a^ω or b^ω. Objective $\overline{W}$ is not DBA-recognizable (a close proof can be found in [BK08, Theorem 4.50]). Still, it is recognizable by a *deterministic coBüchi automaton* similar to the automaton in Figure 8.5, but which accepts infinite words that visit transitions labeled with ● only finitely often. This objective is half-positional, which can be shown using [DJW97, Theorem 6]. However, its prefix classifier has just one state, and there is no way to recognize $\overline{W}$ by building a coBüchi (or even parity) automaton on top of it.

[BK08]: Baier et al. (2008), *Principles of model checking*

[DJW97]: Dziembowski et al. (1997), *How Much Memory is Needed to Win Infinite Games?*

We stress that Condition 3, in its statement, relies on our use of *transition-based* DBAs (i.e., with the acceptance condition defined on the transitions, as opposed to *state-based* DBAs — see Remark 2.7.5). For instance, for $C' \subseteq C$, the objective Büchi(C') satisfies Condition 3. Its prefix classifier has a single state (as it is prefix-independent) and it can indeed be represented by a DBA with a single state: simply, $(\{q_{\text{init}}\}, C, q_{\text{init}}, \delta, B)$ such that $(q_{\text{init}}, c) \in B$ if and only if $c \in C'$. Had we used state-based DBAs, apart from the trivial cases $C' = \emptyset$ or $C' = C$, recognizing Büchi(C') would require two states. This third condition, stated as is, would therefore not apply to this simple half-positional example if we only considered state-based DBAs. For our purposes, considering transition-based DBAs brings more succinct and elegant statements.

This DBA can be simply depicted as

$C \setminus C'$ q_{init} ● C' .
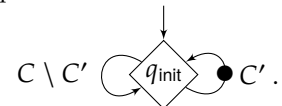
8.3.2 Characterization and corollaries

We have now defined the three conditions required for our characterization.

> **Theorem 8.3.7** (Characterization of half-positional DBA-recognizable objectives) *Let $W \subseteq C^\omega$ be a DBA-recognizable objective. Objective W is half-positional (over all arenas) if and only if*
>
> ▶ *its prefix preorder $\preceq$ is total,*
> ▶ *it is $\mathcal{M}_{\text{triv}}$-progress-consistent, and*
> ▶ *it can be recognized by a Büchi automaton built on top of its prefix classifier $\mathcal{S}_W$.*

Proof. The necessity of the first two conditions (for more general memory structures) was shown in Propositions 7.3.1 and 7.4.4. The necessity of the third condition can be found in Section 8.4, Proposition 8.4.1. The proof of the sufficiency of the conjunction of the three conditions can be found in Section 8.5, Proposition 8.5.4. □

For the use of Proposition 7.3.1, we recall that having a total prefix preorder is equivalent to being $\mathcal{M}_{\text{triv}}$-strongly-monotone.

This characterization is valuable to prove (and disprove) half-positionality of DBA-recognizable objectives. Examples 8.3.1, 8.3.2, and 8.3.4 are *not* half-positional, and each of them falsifies exactly one of the three conditions from the statement. On the other hand, Example 8.3.3 ($W = \text{Büchi}(a) \cup C^* a a C^\omega$) is half-positional. We have already discussed its $\mathcal{M}_{\text{triv}}$-progress-consistency, but it is also straightforward to verify that its prefix preorder is total and that it is recognizable by its prefix classifier: the right congruence has three totally ordered equivalence classes corresponding to the states of the DBA in Figure 8.4.

We state two notable consequences of Theorem 8.3.7 and its proof technique. The first one is the specialization of Theorem 8.3.7 to prefix-independent objectives. It states that all prefix-independent and DBA-recognizable objectives that are half-positional are of the kind $\text{Büchi}(C')$ for some $C' \subseteq C$. Prefix-independence of objectives is a frequent assumption in the literature [Kop06; CN06; GK14; CFGO22] — we show that under this assumption, half-positionality of DBA-recognizable objectives is very easy to understand and characterize.

[Kop06]: Kopczyński (2006), *Half-Positional Determinacy of Infinite Games*
[CN06]: Colcombet et al. (2006), *On the positional determinacy of edge-labeled games*
[GK14]: Gimbert et al. (2014), *Submixing and Shift-Invariant Stochastic Games*
[CFGO22]: Colcombet et al. (2022), *The Theory of Universal Graphs for Infinite Duration Games*

> **Proposition 8.3.8** *Let $W \subseteq C^\omega$ be a prefix-independent, DBA-recognizable objective. Objective W is half-positional if and only if there exists $C' \subseteq C$ such that $W = \text{Büchi}(C')$.*

Proof. The right-to-left implication follows from the established half-positionality of objectives of the kind $\text{Büchi}(C')$ (Theorem 2.6.2). For the left-to-right implication, we assume that W is a prefix-independent, DBA-recognizable, half-positional objective. By Theorem 8.3.7, it is recognized by a DBA $\mathcal{B}$ built on top of $\mathcal{S}_W$. As W is prefix-independent, its prefix classifier has just one state, and there is a single transition from and to this single state for each color. Hence, $W = \text{Büchi}(C')$, where C' is the set of colors whose only transition is a Büchi transition in $\mathcal{B}$. □

Remark 8.3.9 A corollary of this result is that when W is prefix-independent, DBA-recognizable, and half-positional, we also have that $\overline{W}$ is half-positional. Indeed, the complement of objective $W = \text{Büchi}(C')$ is a so-called *coBüchi objective*, which is also known to be half-positional (it is a special case of a parity objective; see Theorem 2.6.3). This statement does not hold in general when W is not prefix-independent, as was shown in Example 8.3.3. Moreover, the reciprocal of the statement also does not hold (there are prefix-independent half-positional objectives recognized by a deterministic *coBüchi* automaton whose complement is not half-positional), as was shown in Example 8.3.6.

Remark 8.3.10 A second corollary is that prefix-independent DBA-recognizable half-positional objectives are closed under finite union (since a finite union of Büchi conditions is a Büchi condition). This settles Kopczyński's conjecture (mentioned in Section 8.1) for DBA-recognizable objectives.

A second consequence of Theorem 8.3.7 and its proof technique shows that half-positionality of DBA-recognizable objectives can be reduced to half-positionality over the restricted class of *finite one-player arenas*. This provides yet a new one-to-two-player lift, with stronger properties than the general ones in Chapters 4 and 5, but for a more restricted class of objectives.

Theorem 8.3.11 (Finite-to-infinite, one-to-two-player lift for half-positional DBAs) *Let $W \subseteq C^\omega$ be a DBA-recognizable objective. If objective W is half-positional over finite one-player arenas, then it is half-positional over all arenas (of any cardinality).*

Proof. We have already shown that for ω-regular objectives, the first two conditions are necessary for half-positionality over finite one-player arenas (Propositions 7.3.1 and 7.4.4). When showing the necessity of the third condition for half-positionality of DBA-recognizable objectives in Section 8.4 (Proposition 8.4.1), we also show its necessity for half-positionality over *finite one-player* arenas. Hence, assuming half-positionality over finite one-player arenas, we have the three conditions from the characterization of Theorem 8.3.7, so we have half-positionality over all arenas. $\square$

8.3.3 Deciding half-positionality in polynomial time

In this section, we assume that C is finite. We show that the problem of deciding, given a DBA $\mathcal{B} = (Q, C, q_{\text{init}}, \delta, B)$ as an input, whether $\mathcal{L}(\mathcal{B})$ is half-positional can be solved in polynomial time, and more precisely in time $\mathcal{O}(|Q|^4 \cdot |C|^2)$.

We investigate how to verify each property used in the characterization of Theorem 8.3.7. Let $\mathcal{B} = (Q, C, q_{\text{init}}, \delta, B)$ be a DBA (we assume w.l.o.g. that all states in Q are reachable from q_{init}) and $W = \mathcal{L}(\mathcal{B})$ be the objective it recognizes. Our algorithm first verifies that the prefix preorder is total and recognizability by $\mathcal{S}_W$, and then, under these first two assumptions, $\mathcal{M}_{\text{triv}}$-progress-consistency. For each condition, we sketch an algorithm to decide it, and we discuss the time complexity of this algorithm.

Total prefix preorder. To check that W has a total prefix preorder, it suffices to check that the states of $\mathcal{B}$ are totally preordered by $\leq_{\mathcal{B}}$. We start by computing, for each pair of states $q, q' \in Q$, whether $q \leq_{\mathcal{B}} q'$, $q' \leq_{\mathcal{B}} q$, or none of these. This can be rephrased as a *containment query* for two DBA-recognizable objectives: if $\mathcal{B}_q = (Q, C, q, \delta, B)$ and $\mathcal{B}_{q'} = (Q, C, q', \delta, B)$, we have that $q \leq_{\mathcal{B}} q'$ if and only if $\mathcal{L}(\mathcal{B}_q) \subseteq \mathcal{L}(\mathcal{B}_{q'})$. Such a problem can be solved in time $\mathcal{O}(|Q|^2 \cdot |C|^2)$ [CDK93]. We can therefore know for all $|Q|^2$ pairs $q, q' \in Q$ whether $q \leq_{\mathcal{B}} q'$, $q' \leq_{\mathcal{B}} q$, $q' \sim_{\mathcal{B}} q$ (as $\sim_{\mathcal{B}} = \leq_{\mathcal{B}} \cap \geq_{\mathcal{B}}$), or none of these in time $\mathcal{O}(|Q|^2 \cdot (|Q|^2 \cdot |C|^2)) = \mathcal{O}(|Q|^4 \cdot |C|^2)$. In particular, the prefix preorder is total if and only if for all $q, q' \in Q$, we have $q \leq_{\mathcal{B}} q'$ or $q' \leq_{\mathcal{B}} q$.

[CDK93]: Clarke et al. (1993), *A Unified Approch for Showing Language Inclusion and Equivalence Between Various Types of omega-Automata*

Recognizability by the prefix classifier. After all the relations $\leq_{\mathcal{B}}$ and $\sim_{\mathcal{B}}$ between pairs of states are computed in the previous step, we can compute the states and transitions of the prefix classifier $\mathcal{S}_W = (Q_W, C, q_{\text{init}}^W, \delta_W)$ by merging all the equivalence classes for $\sim_{\mathcal{B}}$. We assume for simplicity that $Q_W = Q / \sim_{\mathcal{B}}$, and we drop the subscript $\mathcal{B}$ of $\sim_{\mathcal{B}}$ in what follows.

We now wonder whether it is possible to recognize W by carefully selecting a set $B_\sim$ of Büchi transitions in $\mathcal{S}_W$. We simplify the search for such a set with the following result, which shows that *when $\mathcal{B}$ is saturated*, it suffices to try with one specific set $B_\sim$. We can then simply check whether $W = \mathcal{L}((Q_W, C, q_{\text{init}}^W, \delta_W, B_\sim))$, an equivalence query which, as discussed above, can be performed in time $\mathcal{O}(|Q|^2 \cdot |C|^2)$ by checking two containment queries.

> **Lemma 8.3.12** *We assume that $\mathcal{B}$ is saturated and that W is recognized by a DBA built on top of the prefix classifier $\mathcal{S}_W = (Q_W, C, q_{\text{init}}^W, \delta_W)$. We define*
>
> $$B_\sim = \{([q]_\sim, c) \in Q_W \times C \mid \forall q' \in [q]_\sim, (q', c) \in B\}.$$
>
> *Then, W is recognized by $(Q_W, C, q_{\text{init}}^W, \delta_W, B_\sim)$.*

Proof. We assume that W is recognized by a DBA built on top of $\mathcal{S}_W$. We start by saturating this DBA, which yields a set of Büchi transitions B' such that W is also recognized by the saturated DBA $\mathcal{B}' = (Q_W, C, q_{\text{init}}^W, \delta_W, B')$ (Lemma 8.2.3). To prove the claim, we show that $B' = B_\sim$.

We first show that $B' \subseteq B_\sim$. Let $([q]_\sim, c) \notin B_\sim$ — we show that $([q]_\sim, c) \notin B'$. As $([q]_\sim, c) \notin B_\sim$, by definition of $B_\sim$, there is $q' \in Q$ such that $(q', c) \notin B$. As $\mathcal{B}$ is saturated, by Lemma 8.2.4, there exists $w' \in C^*$ such that $cw' \in B\text{-FreeCycles}_\mathcal{B}(q')$. By construction of the prefix classifier, $\delta_\sim^*([q]_\sim, cw') = [q]_\sim$. Also, as $W = \mathcal{L}(\mathcal{B}')$, word $(cw')^\omega$ must be rejected from $[q]_\sim$ in $\mathcal{B}'$. Therefore, $([q]_\sim, c)$ cannot be a Büchi transition of $\mathcal{B}'$ and is not in B'.

We now show that $B_\sim \subseteq B'$. Let $([q]_\sim, c) \notin B'$ — we show that $([q]_\sim, c) \notin B_\sim$. As $\mathcal{B}'$ is saturated, by Lemma 8.2.4, there exists $w' \in C^*$ such that $cw' \in B\text{-FreeCycles}_{\mathcal{B}'}([q]_\sim)$. As $W = \mathcal{L}(\mathcal{B}')$, word $(cw')^\omega$ is rejected from any state in $[q]_\sim$ in $\mathcal{B}$. If for all $q' \in [q]_\sim$, (q', c) was in B, $(cw')^\omega$ would be accepted from all states in $[q]_\sim$ in $\mathcal{B}$. Hence, there exists $q' \in [q]_\sim$ such that $(q', c) \notin B$. We conclude that $([q]_\sim, c) \notin B_\sim$. $\qquad\square$

$\mathcal{M}_{\mathsf{triv}}$**-progress-consistency.** We assume that we have already checked that W is recognizable by a Büchi automaton built on top of $\mathcal{S}_W$, and that we know the (total) ordering of the states. We show that checking $\mathcal{M}_{\mathsf{triv}}$-progress-consistency, under these two hypotheses, can be done in polynomial time. We prove a lemma reducing the search for words witnessing that W is not $\mathcal{M}_{\mathsf{triv}}$-progress-consistent to a problem computationally easier to investigate. The core idea is the same as in the proof for regular objectives (Lemma 7.6.3): we just slightly adapt the property to the Büchi acceptance condition.

> **Lemma 8.3.13** *We assume that $\mathcal{B}$ is built on top of the prefix classifier $\mathcal{S}_W$ and that the prefix preorder of W is total. Then, W is $\mathcal{M}_{\mathsf{triv}}$-progress-consistent if and only if for all $q, q' \in Q$ with $q <_\mathcal{B} q'$,*
>
> $$\{w \in C^+ \mid \delta^*(q, w) = q'\} \cap B\text{-FreeCycles}_\mathcal{B}(q') = \emptyset.$$

Proof. For the left-to-right implication, we assume by contrapositive that there exist $q, q' \in Q$ with $q <_\mathcal{B} q'$ and $w \in C^+$ such that $\delta^*(q, w) = q'$ and $w \in B\text{-FreeCycles}_\mathcal{B}(q')$. Let $w_q \in C^*$ be a word such that $\delta^*(q_{\mathsf{init}}, w_q) = q$. We have that $w_q \prec w_q w$, but $w_q w^\omega$ is not accepted by $\mathcal{B}$ as w is a cycle on q' that does not see any Büchi transition. Hence, W is not $\mathcal{M}_{\mathsf{triv}}$-progress-consistent.

For the right-to-left implication, we assume by contrapositive that W is not $\mathcal{M}_{\mathsf{triv}}$-progress-consistent. Thus, there exist $w' \in C^*$ and $w \in C^+$ such that $w' \prec w'w$ and $w'w^\omega \notin W$. Let $q_1 = \delta^*(q_{\mathsf{init}}, w')$ and $q_2 = \delta^*(q_1, w)$ — we have $q_1 \prec q_2$. As $q_1 \prec q_2$, by Lemma 2.8.10, we have $\delta^*(q_1, w) = q_2 \preceq \delta^*(q_2, w)$. We distinguish two cases, using the fact that there is exactly one state per equivalence class for $\sim_\mathcal{B}$. We represent what happens in Figure 8.6.

- ▶ If $q_2 = \delta^*(q_2, w)$, we then have that $w \in B\text{-FreeCycles}_\mathcal{B}(q_2)$, and we have what we want with $q = q_1$ and $q' = q_2$.
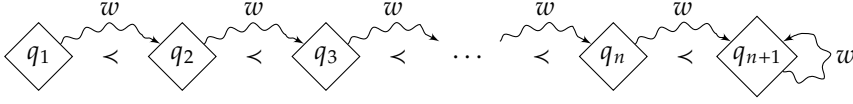
Figure 8.6: Situation in the proof of Lemma 8.3.13.

▶ If not, we have that $q_2 \prec \delta^*(q_2, w)$. Let $q_3 = \delta^*(q_2, w)$. We can repeat the argument on q_2 and q_3: either $w \in B\text{-FreeCycles}_{\mathcal{B}}(q_3)$ and we are done, or $q_3 \prec \delta^*(q_3, w)$. As there are finitely many states, this process necessarily ends with two states $q = q_n$ and $q' = q_{n+1}$ such that $\delta^*(q, w) = q'$ and $w \in B\text{-FreeCycles}_{\mathcal{B}}(q')$. □

Notice that for each pair of states $q, q' \in Q$, the sets $\{w \in C^+ \mid \delta(q, w) = q'\}$ and $B\text{-FreeCycles}_{\mathcal{B}}(q')$ are both regular languages recognized by deterministic finite automata with at most $|Q|$ states. The emptiness of their intersection can be decided in time $\mathcal{O}(|Q|^2 \cdot |C|)$ (by solving a reachability problem in the product of the two automata). Thanks to Lemma 8.3.13, we can therefore decide whether $\mathcal{B}$ is $\mathcal{M}_{\text{triv}}$-progress-consistent in time $\mathcal{O}(|Q|^2 \cdot (|Q|^2 \cdot |C|)) = \mathcal{O}(|Q|^4 \cdot |C|)$: for all $|Q|^2$ pairs of states $q, q' \in Q$, if $q \prec q'$, we test the emptiness of the intersection of these two deterministic finite automata.

Complexity wrap-up. By checking the three conditions as explained and in this order, the time complexities are respectively $\mathcal{O}(|Q|^4 \cdot |C|^2)$, $\mathcal{O}(|Q|^2 \cdot |C|^2)$, and $\mathcal{O}(|Q|^4 \cdot |C|)$. This yields a time complexity of $\mathcal{O}(|Q|^4 \cdot |C|^2)$ for the whole algorithm.

8.4 Necessity of the third condition

We have already shown that having a total preorder (i.e., $\mathcal{M}_{\text{triv}}$-strong-monotony) and $\mathcal{M}_{\text{triv}}$-progress-consistency are necessary conditions for half-positionality over finite one-player arenas, through more general proofs for ω-regular objectives in Chapter 7 (Propositions 7.3.1 and 7.4.4).

We prove the necessity of the third condition: for a DBA-recognizable objective, being recognized by a Büchi automaton built on top of its prefix classifier $\mathcal{S}_W$ is necessary for half-positionality. This section is devoted to the proof of this result, which is more involved than the proofs for the two other conditions.

Proposition 8.4.1 (Necessity of the recognizability by the prefix classifier)
Let $W \subseteq C^\omega$ be a DBA-recognizable objective that is half-positional over finite one-player arenas. Then, W is recognized by a Büchi automaton built on top of $\mathcal{S}_W$.

We fix an objective $W \subseteq C^\omega$ recognized by a DBA $\mathcal{B} = (Q, C, q_{\text{init}}, \delta, B)$. We make the assumption that W **is half-positional over finite one-player arenas**. Our goal is to show that W can be defined by a Büchi automaton

built on top of $\mathcal{S}_W$. We assume w.l.o.g. that $\mathcal{B}$ **is saturated**. Many upcoming arguments heavily rely on this assumption through the use of Lemma 8.2.4 (any B-free word can be completed into a B-free cycle).

Our proof first assumes in Subsection 8.4.1 that $\mathcal{B}$ recognizes a prefix-independent objective. We will then build on this first case to conclude for the general case in Subsection 8.4.2. We provide a proof sketch at the start of each subsection.

8.4.1 Prefix-independent case

We assume that the **objective** W **recognized by** $\mathcal{B}$ **is prefix-independent**, so all the states of $\mathcal{B}$ are equivalent for $\sim$. We want to show that W can be recognized by a Büchi automaton built on top of $\mathcal{S}_W$, and in this case, the automaton structure $\mathcal{S}_W$ has just one state. Therefore, we want to find $C' \subseteq C$ such that $W = \mathsf{Büchi}(C')$. We start with a high level description of the proof technique.

Sketch of proof. The goal is to find a suitable definition for C'. To do so, we exhibit a state $q_{\max}$ of $\mathcal{B}$ that is "the most rejecting state of the automaton": it satisfies that the set of B-free words from $q_{\max}$ contains the B-free words from all the other states ($q_{\max}$ is then called a *B-free-maximum*) *and* that the set of B-free cycles on $q_{\max}$ contains the B-free cycles on all the other states (it is also a *B-free-cycle-maximum*). We define C' as the set of colors $c \in C$ such that $(q_{\max}, c) \in B$.

We first show that if a B-free-maximum exists, we can assume w.l.o.g. that it is unique (Lemma 8.4.2). In Lemmas 8.4.4, 8.4.5, and 8.4.6, we show the existence of a B-free-cycle-maximum. This part of the proof relies on the half-positionality over finite one-player arenas of W. Finally, defining C' using $q_{\max}$ as described above, we prove that $W = \mathsf{Büchi}(C')$ (Lemma 8.4.7). □

We call a state $q_{\max} \in Q$ of $\mathcal{B}$ a *B-free-maximum* (resp. a *B-free-cycle-maximum*) if for all $q \in Q$, we have $B\text{-}\mathsf{Free}_{\mathcal{B}}(q) \subseteq B\text{-}\mathsf{Free}_{\mathcal{B}}(q_{\max})$ (resp. $B\text{-}\mathsf{FreeCycles}_{\mathcal{B}}(q) \subseteq B\text{-}\mathsf{FreeCycles}_{\mathcal{B}}(q_{\max})$). We remark that if $\mathcal{B}$ is saturated, a B-free-cycle-maximum is also a B-free-maximum (this can be shown using Lemma 8.2.4).

We first show that we can remove states from $\mathcal{B}$, while still recognizing the same objective, until it has at most one B-free-maximum.

Lemma 8.4.2 *There exists a DBA $\mathcal{B}'$ recognizing W with at most one B-free-maximum.*

Proof. Assume that $q_{max}^1, q_{max}^2 \in Q$ are distinct B-free-maxima. In particular, $B\text{-Free}_{\mathcal{B}}(q_{max}^1) = B\text{-Free}_{\mathcal{B}}(q_{max}^2)$. We show that in such a situation, the objective recognized by $\mathcal{B}$ can be recognized by an automaton with one less state, in which we discard one of the two B-free-maxima. To simplify the upcoming arguments, we assume that $q_{init} = q_{max}^1$ (which is without loss of generality as W is prefix-independent and all states of $\mathcal{B}$ are equivalent for $\sim$).

We define a new automaton in which we remove q_{max}^2 and redirect all its incoming transitions to q_{max}^1. Formally, let $\mathcal{B}' = (Q', C, q'_{init}, \delta', B')$ be such that

- $Q' = Q \setminus \{q_{max}^2\}$, $q'_{init} = q_{max}^1$,
- for $q \in Q'$ and $c \in C$, if $\delta(q, c) = q_{max}^2$, then $\delta'(q, c) = q_{max}^1$; otherwise, $\delta'(q, c) = \delta(q, c)$,
- for $q \in Q'$ and $c \in C$, $(q, c) \in B'$ if and only if $(q, c) \in B$.

We also assume that states that are not reachable from q'_{init} in $\mathcal{B}'$ are removed from Q'.

We show that this automaton with (at least) one less state recognizes the same objective as $\mathcal{B}$. Let $w = c_1 c_2 \ldots \in C^\omega$ be an infinite word. We show that w is accepted by $\mathcal{B}$ if and only if it is accepted by $\mathcal{B}'$.

Let $\Delta_{\not\mapsto q_{max}^2} = \{(q, c) \in Q' \times C \mid \delta(q, c) = q_{max}^2\}$ be the transitions of $\mathcal{B}'$ that were directed to q_{max}^2 in $\mathcal{B}$, but that are now redirected to q_{max}^1 in $\mathcal{B}'$. Let ϱ be the run of $\mathcal{B}$ on w, and $\varrho' = (q'_0, c_1)(q'_1, c_2) \ldots$ be the run of $\mathcal{B}'$ on w. The two runs start by taking corresponding transitions, but differ once a transition in $\Delta_{\not\mapsto q_{max}^2}$ is taken.

We first assume that ϱ' uses transitions in $\Delta_{\not\mapsto q_{max}^2}$ only finitely many times. Then, there exists $k \geq 0$ such that $q'_k = q_{max}^1$ and for all $l \geq k$, $(q'_l, c_{l+1}) \notin \Delta_{\not\mapsto q_{max}^2}$. Let $w_{>k} = c_{k+1} c_{k+2} \ldots$ be the infinite word consisting of the colors taken after the last occurrence of a transition in $\Delta_{\not\mapsto q_{max}^2}$. We have that

$\mathcal{B}$ accepts w

$\Longleftrightarrow \mathcal{B}$ accepts $w_{>k}$ as $\mathcal{B}$ recognizes a prefix-independent objective

$\Longleftrightarrow \mathcal{B}'$ accepts $w_{>k}$ as $w_{>k}$ visits exactly the same transitions as in $\mathcal{B}$

$\Longleftrightarrow \mathcal{B}'$ accepts w as $c_1 \ldots c_k$ is a cycle on the initial state q_{max}^1 of $\mathcal{B}'$.

We now assume that ϱ' uses transitions in $\Delta_{\not\mapsto q_{max}^2}$ infinitely many times. We decompose ϱ' into infinitely many finite runs $\varrho'_1, \varrho'_2, \ldots$ such that $\varrho' = \varrho'_1 \varrho'_2 \ldots$ and every run ϱ'_i sees exactly one transition in $\Delta_{\not\mapsto q_{max}^2}$ as its last transition. This implies that all these finite runs start in state q_{max}^1. We represent run ϱ' in Figure 8.7. We define words $w_1, w_2, \ldots$ as the respective projection of runs $\varrho'_1, \varrho'_2, \ldots$ to their colors (we have $w = w_1 w_2 \ldots$). Notice

$$\varrho': \quad \underset{\Delta_{\nrightarrow q_2}}{\overset{q^1_{\max}\quad w_1\quad q^1_{\max}}{\rightsquigarrow}} \quad \underset{\Delta_{\nrightarrow q_2}}{\overset{w_2\quad q^1_{\max}}{\rightsquigarrow}} \quad \underset{\Delta_{\nrightarrow q_2}}{\overset{w_3\quad q^1_{\max}}{\rightsquigarrow}} \quad \cdots$$
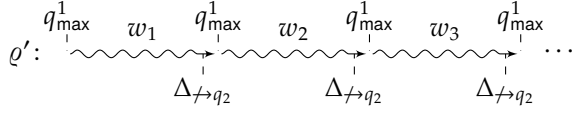
Figure 8.7: Lemma 8.4.2: features of run ϱ' when it takes infinitely many transitions in $\Delta_{\nrightarrow q^2_{\max}}$.

that

$$\forall i \geq 1, w_i \in B\text{-Free}_{\mathcal{B}}(q^1_{\max}) \Longleftrightarrow w_i \in B\text{-Free}_{\mathcal{B}'}(q^1_{\max}), \qquad (8.1)$$

as the transitions used by w_i from $q^1_{\max}$ in $\mathcal{B}'$ correspond to the transitions used by w_i from $q^1_{\max}$ in $\mathcal{B}$ (this property is not necessarily true for all words, but is true for these words that read only one transition in $\Delta_{\nrightarrow q^2_{\max}}$ as their last transition). We also have by construction that

$$\forall i \geq 1, \delta^*(q^1_{\max}, w_i) = q^2_{\max}. \qquad (8.2)$$

We distinguish whether w is accepted or rejected by $\mathcal{B}'$.

Assume w is accepted by $\mathcal{B}'$. Then, we know that for infinitely many $i \in \mathbb{N}$, $w_i \notin B\text{-Free}_{\mathcal{B}'}(q^1_{\max})$. This implies that for these indices i, $w_i \notin B\text{-Free}_{\mathcal{B}}(q^1_{\max})$ by Equation (8.1). As $q^1_{\max}$ is a B-free-maximum, for all $q \in Q$, $w_i \notin B\text{-Free}_{\mathcal{B}}(q)$ (this is simply the contrapositive of the definition of B-free-maximum). Hence, for infinitely many $i \in \mathbb{N}$, when w_i is read in $\mathcal{B}$ (no matter from where), a Büchi transition is seen, so w is accepted by $\mathcal{B}$.

Assume w is rejected by $\mathcal{B}'$. Then there exists $k \geq 0$ such that for all $l \geq k$, $w_l \in B\text{-Free}_{\mathcal{B}'}(q^1_{\max})$. As $\mathcal{B}$ is prefix-independent, up to removing the start of w, we assume w.l.o.g. that $k = 1$. We show by induction that

$$\forall i \geq 1, B\text{-Free}_{\mathcal{B}}(q^1_{\max}) = B\text{-Free}_{\mathcal{B}}(\delta^*(q^1_{\max}, w_1 \ldots w_i)).$$

This is true for $i = 1$, as $\delta^*(q^1_{\max}, w_1) = q^2_{\max}$ by Equation (8.2) and the fact that $q^1_{\max}$ and $q^2_{\max}$ are both B-free-maxima. Assume $B\text{-Free}_{\mathcal{B}}(q^1_{\max}) = B\text{-Free}_{\mathcal{B}}(\delta^*(q^1_{\max}, w_1 \ldots w_{i-1}))$ for some $i \geq 2$. Then, by Lemma 8.2.1, as $w_i \in B\text{-Free}_{\mathcal{B}}(q^1_{\max})$, we have

$$B\text{-Free}_{\mathcal{B}}(\delta^*(q^1_{\max}, w_i)) = B\text{-Free}_{\mathcal{B}}(\delta^*(q^1_{\max}, w_1 \ldots w_{i-1}w_i)).$$

Equation (8.2) gives $B\text{-Free}_{\mathcal{B}}(\delta^*(q^1_{\max}, w_i)) = B\text{-Free}_{\mathcal{B}}(q^2_{\max})$, which is itself equal to $B\text{-Free}_{\mathcal{B}}(q^1_{\max})$. We now know that for all $i \geq 1$, $w_i \in B\text{-Free}_{\mathcal{B}}(q^1_{\max})$ by Equation (8.1). Therefore, we conclude that for all $i \geq 1$, $w_i \in B\text{-Free}_{\mathcal{B}}(\delta^*(q^1_{\max}, w_1 \ldots w_{i-1}))$. In particular, w sees no Büchi transition when read from $q^1_{\max}$ in $\mathcal{B}$ and is also rejected by $\mathcal{B}$.

We have shown that $\mathcal{B}'$ is a DBA with fewer states than $\mathcal{B}$ recognizing W. If $\mathcal{B}'$ still has two or more B-free-maxima, we repeat our construction until there is at most one left. $\qquad \square$

Thanks to Lemma 8.4.2, we now assume w.l.o.g. that $\mathcal{B}$ has at most one B-free-maximum. We intend to show that there exists a B-free-cycle-

maximum. To do so, we exhibit an (infinite) arena in which $\mathcal{P}_1$ has no winning strategy, which we prove by using half-positionality of W over finite one-player arenas. We then prove that the non-existence of a *B*-free-cycle-maximum would imply that $\mathcal{P}_1$ has a winning strategy in this arena (Lemma 8.4.6).
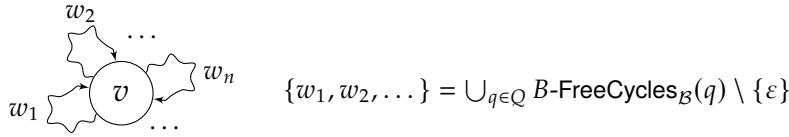
Let $\mathcal{A}_\mathcal{B}$ be the infinite one-player arena of $\mathcal{P}_1$ depicted in Figure 8.8. This arena consists of one vertex v with a choice to make among all non-empty words that are *B*-free cycles from some state of $\mathcal{B}$. Vertex v is the only vertex with multiple outgoing edges. The goal of the next three short lemmas is to show that in this arena, $\mathcal{P}_1$ has no winning strategy.

> **Lemma 8.4.3** *If $\mathcal{P}_1$ has a winning strategy in $\mathcal{A}_\mathcal{B}$, then $\mathcal{P}_1$ has a memoryless winning strategy.*

Proof. Suppose that there is a winning strategy of $\mathcal{P}_1$ in $\mathcal{A}_\mathcal{B}$. Let $w = w_1 w_2 \ldots$ be an infinite winning word such that for $i \geq 1$,

$$w_i \in \bigcup_{q \in Q} B\text{-FreeCycles}_\mathcal{B}(q) \setminus \{\varepsilon\}.$$

Let $q_0 = q_{\text{init}}$, and for $i \geq 1$, let $q_i = \delta^*(q_{\text{init}}, w_1 \ldots w_i)$ be the current automaton state after reading the first i finite words composing w. As there are only finitely many automaton states and w is winning, there are $k, l \geq 1$ with $k < l$ such that $q_k = q_l$ and $w_{k+1} \ldots w_l \notin B\text{-Free}_\mathcal{B}(q_k)$. Word $w_1 \ldots w_k (w_{k+1} \ldots w_l)^\omega$ is also a winning word and uses only finitely many different words in $\bigcup_{q \in Q} B\text{-FreeCycles}_\mathcal{B}(q) \setminus \{\varepsilon\}$.

Hence, there is a finite restriction ("subarena") $\mathcal{A}'_\mathcal{B}$ of the arena $\mathcal{A}_\mathcal{B}$ with at most l choices in v in which $\mathcal{P}_1$ has a winning strategy. Arena $\mathcal{A}'_\mathcal{B}$ being finite and one-player, half-positionality of W over finite one-player arenas implies that $\mathcal{P}_1$ has a memoryless winning strategy in $\mathcal{A}'_\mathcal{B}$. This memoryless winning strategy can also be played in $\mathcal{A}_\mathcal{B}$ (as every choice available in $\mathcal{A}'_\mathcal{B}$ is also available in $\mathcal{A}_\mathcal{B}$). $\square$

> **Lemma 8.4.4** *No memoryless strategy of $\mathcal{P}_1$ is winning in $\mathcal{A}_\mathcal{B}$.*

Proof. Any memoryless strategy of $\mathcal{P}_1$ generates a word w^ω, where $w \in B\text{-FreeCycles}_\mathcal{B}(q) \setminus \{\varepsilon\}$ for some $q \in Q$. In particular, word w^ω is rejected when it is read from state q. As all the states in Q are equivalent for $\sim$ (as we assume that W is prefix-independent), we have $q \sim q_{\text{init}}$, so w^ω is also rejected when read from the initial state q_{init} of the automaton. $\square$

Combining Lemmas 8.4.3 and 8.4.4, we deduce the desired result.

> **Lemma 8.4.5** *No strategy of $\mathcal{P}_1$ is winning in $\mathcal{A}_{\mathcal{B}}$.*

We use the statement of Lemma 8.4.5 to show the existence of a *B-free-cycle-maximum*.

> **Lemma 8.4.6** *There exists a B-free-maximum q_{max} in $\mathcal{B}$ that is a moreover a B-free-cycle-maximum: for all $q \in Q$, B-FreeCycles$_{\mathcal{B}}(q) \subseteq$ B-FreeCycles$_{\mathcal{B}}(q_{max})$.*

Proof. Let us assume by contradiction that there is no *B-free-maximum*, or if there is one, that it is not a *B-free-cycle-maximum*. We show how to build a winning strategy of $\mathcal{P}_1$ in $\mathcal{A}_{\mathcal{B}}$, contradicting Lemma 8.4.5. To do so, we build an infinite word accepted by $\mathcal{B}$ by combining finite words that are *B-free cycles* from some state.

We claim that for $q \in Q$ which is not a *B-free-maximum*, there exists

$$w_q \in \bigcup_{q' \in Q} B\text{-FreeCycles}_{\mathcal{B}}(q') \setminus \{\varepsilon\} \text{ such that } w_q \notin B\text{-Free}_{\mathcal{B}}(q).$$

Let $q' \in Q$ be such that $B\text{-Free}_{\mathcal{B}}(q') \not\subseteq B\text{-Free}_{\mathcal{B}}(q)$, which exists as q is not a *B-free-maximum*. Let $w_1 \in B\text{-Free}_{\mathcal{B}}(q') \setminus B\text{-Free}_{\mathcal{B}}(q)$. By Lemma 8.2.4, there exists $w_2 \in C^*$ such that $w_1 w_2 \in B\text{-FreeCycles}_{\mathcal{B}}(q')$ (this holds as we have assumed w.l.o.g. that $\mathcal{B}$ is saturated). As $w_1 \notin B\text{-Free}_{\mathcal{B}}(q)$, we also have $w_1 w_2 \notin B\text{-Free}_{\mathcal{B}}(q)$. Taking $w_q = w_1 w_2$ proves the claim. For $q \in Q$ not a *B-free-maximum*, we fix $w_q \in C^+$ such that $w_q \in \bigcup_{q' \in Q} B\text{-FreeCycles}_{\mathcal{B}}(q') \setminus \{\varepsilon\}$ and $w_q \notin B\text{-Free}_{\mathcal{B}}(q)$.

Let $q_{max} \in Q$ be a *B-free-maximum* (that we suppose to be unique by Lemma 8.4.2), if it exists. We suppose by contradiction that q_{max} is not a *B-free-cycle-maximum*: there is $q \in Q$ such that $B\text{-FreeCycles}_{\mathcal{B}}(q) \not\subseteq B\text{-FreeCycles}_{\mathcal{B}}(q_{max})$. Let $w_{max} \in B\text{-FreeCycles}_{\mathcal{B}}(q) \setminus B\text{-FreeCycles}_{\mathcal{B}}(q_{max})$. Notice that as $w_{max} \in B\text{-Free}_{\mathcal{B}}(q)$ and q_{max} is a *B-free-maximum*, $w_{max} \in B\text{-Free}_{\mathcal{B}}(q_{max})$. Therefore, w_{max} cannot be a cycle on q_{max}, i.e.,

$$\delta^*(q_{max}, w_{max}) \neq q_{max}.$$

We build iteratively an infinite winning word that can be played by $\mathcal{P}_1$ in $\mathcal{A}_{\mathcal{B}}$. As $\mathcal{P}_1$ plays, we keep track in parallel of the current automaton state. The game starts in v, with current automaton state $q_0 = q_{init}$. Let $n \geq 0$. We distinguish two cases.

▶ If q_n is not a *B-free-maximum*, then $\mathcal{P}_1$ plays word w_{q_n}. As $w_{q_n} \notin B\text{-Free}_{\mathcal{B}}(q_n)$, a Büchi transition is seen along the way. The current automaton state becomes $q_{n+1} = \delta^*(q_n, w_{q_n})$.

▶ If $q_n = q_{max}$ is the B-free-maximum, then $\mathcal{P}_1$ plays w_{max}. The current automaton state becomes $q_{n+1} = \delta(q_{max}, w_{max})$, which is not equal to q_{max}.

For infinitely many $i \geq 0$, the automaton state q_i is not a B-free-maximum (as there is at most one B-free-maximum in $\mathcal{B}$, and it cannot appear twice in a row). Therefore, we have described a winning strategy for $\mathcal{P}_1$, since the corresponding run over $\mathcal{B}$ visits infinitely often a Büchi transition. □

We now know that there exists a **unique B-free-maximum** $q_{max} \in Q$, and moreover, that q_{max} **is a B-free-cycle-maximum**. We show how to use the outgoing transitions of q_{max} in order to realize W as an objective of the kind Büchi(C') for some $C' \subseteq C$, which is the goal of the current subsection.

Lemma 8.4.7 *There exists $C' \subseteq C$ such that $W = Büchi(C')$.*

Proof. Let $C' = \{c \in C \mid (q_{max}, c) \in B\}$ — equivalently, if we consider colors as words with one letter, C' is the set of colors c such that $c \notin B\text{-Free}_\mathcal{B}(q_{max})$.

We first show that Büchi(C') $\subseteq W$. Let $c \in C'$. Then, $c \notin B\text{-Free}_\mathcal{B}(q_{max})$. As q_{max} is a B-free-maximum, for all $q \in Q$, $c \notin B\text{-Free}_\mathcal{B}(q)$. Therefore, any word seeing infinitely many colors in C' sees infinitely many Büchi transitions and is accepted by $\mathcal{B}$.

We now show that $W \subseteq$ Büchi(C'). By contrapositive, let $w = c_1 c_2 \ldots \notin$ Büchi(C') be an infinite word with only finitely many colors in C'. We show that $w \notin W$. As W is prefix-independent, we may assume w.l.o.g. that w has no color in C', i.e., that for all $i \geq 1$, $c_i \in C \setminus C'$. We claim that when read from q_{max}, word w sees no Büchi transition and is thus rejected. This implies that $w \notin W$ as $q_{max} \sim q_{init}$.

Assume by contradiction that there is some Büchi transition when reading w from q_{max}, i.e., there exists $k \geq 0$ such that for $w_{\leq k} = c_1 \ldots c_k$, $w_{\leq k} \in B\text{-Free}_\mathcal{B}(q_{max})$, but $w_{\leq k} c_{k+1} \notin B\text{-Free}_\mathcal{B}(q_{max})$. We will deduce that (q_{max}, c_{k+1}) is a Büchi transition, contradicting that $c_{k+1} \in C \setminus C'$.

We depict the situation in Figure 8.9. Let $q_1 = \delta^*(q_{max}, w_{\leq k})$ (whether q_1 equals q_{max} or not does not matter). By Lemma 8.2.4, there exists $w' \in C^*$ such that $w_{\leq k} w' \in B\text{-FreeCycles}_\mathcal{B}(q_{max})$. By construction, we have $w' w_{\leq k} \in B\text{-FreeCycles}_\mathcal{B}(q_1)$. As q_{max} is a B-free-cycle-maximum, we have $B\text{-FreeCycles}_\mathcal{B}(q_1) \subseteq B\text{-FreeCycles}_\mathcal{B}(q_{max})$, so we also have $w' w_{\leq k} \in$
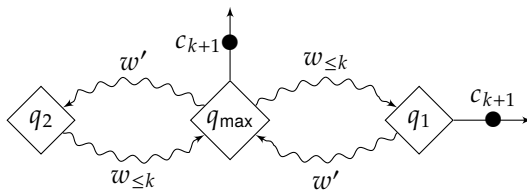
Figure 8.9: Situation in the proof of Lemma 8.4.7, with $w_{\leq k} \in B\text{-Free}_\mathcal{B}(q_{max})$ but $w_{\leq k} c_{k+1} \notin B\text{-Free}_\mathcal{B}(q_{max})$.

B-FreeCycles$_\mathcal{B}(q_{max})$. Let $q_2 = \delta^*(q_{max}, w')$. Notice that $q_{max} = \delta^*(q_2, w_{\leq k})$ and $w_{\leq k} \in B$-Free$_\mathcal{B}(q_2)$. As q_{max} is a B-free-maximum and $w_{\leq k}c_{k+1} \notin B$-Free$_\mathcal{B}(q_{max})$, we also have that $w_{\leq k}c_{k+1} \notin B$-Free$_\mathcal{B}(q_2)$. Therefore, transition (q_{max}, c_{k+1}) must be a Büchi transition, which contradicts that $c_{k+1} \in C \setminus C'$. □

8.4.2 General case

We now relax the prefix-independence assumption on W. We still assume that W **is half-positional over finite one-player arenas**, and show that W can be recognized by a Büchi automaton built on top of $\mathcal{S}_W$. If $\mathcal{B}$ has exactly one state per equivalence class for $\sim$, it means that it is built on top of $\mathcal{S}_W$, and we are done. If not, **let** $q_\sim \in Q$ **be a state such that** $|[q_\sim]_\sim| \geq 2$.

We briefly sketch the proof technique for this section.

Sketch of proof. Our proof will show how to modify $\mathcal{B}$ by "merging" all states in equivalence class $[q_\sim]_\sim$ into a single state, while still recognizing the same objective W. The main technical argument is to build a variant $W_{[q_\sim]}$ of objective W on a new set of colors $C_{[q_\sim]}$, that turns out to also be half-positional over finite one-player arenas and DBA-recognizable, but which is *prefix-independent*. We can therefore use Lemma 8.4.7 from Subsection 8.4.1 and find $C'_{[q_\sim]} \subseteq C_{[q_\sim]}$ such that $W_{[q_\sim]} = \mathsf{Büchi}(C'_{[q_\sim]})$. Then, we exhibit a state $q_{max} \in [q_\sim]_\sim$ whose B-free words are tightly linked to the elements of $C'_{[q_\sim]}$ (Lemma 8.4.8 and Corollary 8.4.9). Finally, akin to the way we removed a B-free-maximum in Lemma 8.4.2, we show that it is still possible to recognize W while keeping only state q_{max} in $[q_\sim]_\sim$ (Lemma 8.4.10).

Once we know how to merge the equivalence class $[q_\sim]_\sim$ into a single state, we can simply repeat the operation for each equivalence class with multiple states, until we obtain a DBA built on top of $\mathcal{S}_W$. □

We define a new set of colors $C_{[q_\sim]}$ using finite words in C^+ such that

$$C_{[q_\sim]} = \{w \in C^+ \mid \delta^*(q_\sim, w) \sim q_\sim\}.$$

This set contains all the finite words that, read from $q_\sim$, come back to a state in $[q_\sim]_\sim$. By Lemma 2.8.10, for all $q \in [q_\sim]_\sim$, for all $w \in C_{[q_\sim]}$, we also have that $\delta^*(q, w) \sim \delta^*(q_\sim, w) \sim q_\sim$. The set $C_{[q_\sim]}$ therefore corresponds to the set of words with the seemingly stronger property that, when read from any state in $[q_\sim]_\sim$, come back to a state in $[q_\sim]_\sim$. We define an objective $W_{[q_\sim]}$ of infinite words on this new set of colors such that

$$W_{[q_\sim]} = \{w_1 w_2 \ldots \in C_{[q_\sim]}^\omega \mid w_1 w_2 \ldots \in q_\sim^{-1} W\}.$$

We show that $W_{[q_\sim]}$ has the three conditions allowing us to apply Lemma 8.4.7 to it.

- ▶ Objective $W_{[q_\sim]}$ is DBA-recognizable: we consider the DBA $\mathcal{B}_{[q_\sim]} = ([q_\sim]_\sim, C_{[q_\sim]}, q_\sim, \delta', B')$, whose update function δ' is the restriction of δ^* to $[q_\sim]_\sim \times C_{[q_\sim]}$, and $B' = \{(q, w) \in [q_\sim]_\sim \times C_{[q_\sim]} \mid w \notin B\text{-Free}_{\mathcal{B}}(q)\}$.
- ▶ Objective $W_{[q_\sim]}$ is prefix-independent, as adding or removing a finite number of cycles on $q_\sim$ does not affect the accepted status of a word in $q_\sim^{-1}W$.
- ▶ Half-positionality of $W_{[q_\sim]}$ over finite one-player arenas is implied by half-positionality of W over finite one-player arenas. Indeed, every (finite one-player) arena $\mathcal{A}_{[q_\sim]}$ using colors in $C_{[q_\sim]}$ can be transformed into a (finite one-player) arena $\mathcal{A}$ with similar properties using colors in C. Two transformations are applied: (*i*) we replace every $C_{[q_\sim]}$-colored edge in $\mathcal{A}_{[q_\sim]}$ by a corresponding finite chain of C-colored edges, and (*ii*) for every vertex v of $\mathcal{A}_{[q_\sim]}$, we prefix it with a chain of C-colored edges starting from a vertex v' reading a word $w_{q_\sim}$ such that $\delta^*(q_{\text{init}}, w_{q_\sim}) = q_\sim$. We then have that $\mathcal{P}_1$ has a winning strategy from a vertex v in $\mathcal{A}_{[q_\sim]}$ if and only if $\mathcal{P}_1$ has a winning strategy from v' in $\mathcal{A}$, and a memoryless winning strategy from v' in $\mathcal{A}$ can be transformed into a memoryless winning strategy from v in $\mathcal{A}_{[q_\sim]}$.

By Lemma 8.4.7, there exists a set $C'_{[q_\sim]} \subseteq C_{[q_\sim]}$ such that

$$W_{[q_\sim]} = \text{Büchi}(C'_{[q_\sim]}).$$

We now show links between B-free words from states of $[q_\sim]_\sim$ and the words in $C'_{[q_\sim]}$. The arguments once again rely on the saturation of $\mathcal{B}$.

Lemma 8.4.8

- ▶ *Let $w \in C_{[q_\sim]}$. If $w \in C'_{[q_\sim]}$, then for all $q \in [q_\sim]_\sim$, $w \notin B\text{-Free}_{\mathcal{B}}(q)$.*
- ▶ *There exists $q \in [q_\sim]_\sim$ such that, for all $w \in C_{[q_\sim]} \setminus C'_{[q_\sim]}$, $w \in B\text{-Free}_{\mathcal{B}}(q)$.*

Proof. For the first item, we assume by contrapositive that there exists a state $q \in [q_\sim]_\sim$ such that $w \in B\text{-Free}_{\mathcal{B}}(q)$. By Lemma 8.2.4, there is $w' \in C^*$ such that $ww' \in B\text{-FreeCycles}_{\mathcal{B}}(q)$. In particular, $(ww')^\omega \notin q^{-1}W = q_\sim^{-1}W$. We can assume w.l.o.g. that $w' \in C^+$ (if $w' = \varepsilon$, then we can simply replace it with $w' = w$). Therefore, $(ww')^\omega$ is also an infinite word on $C_{[q_\sim]}$, and we have $(ww')^\omega \notin W_{[q_\sim]}$ since $(ww')^\omega \notin q_\sim^{-1}W$. As $W_{[q_\sim]} = \text{Büchi}(C'_{[q_\sim]})$, clearly $w \notin C'_{[q_\sim]}$, which ends the proof of the first item.

For the second item, assume by contradiction that for all $q \in [q_\sim]_\sim$, there exists $w_q \in C_{[q_\sim]} \setminus C'_{[q_\sim]}$ such that $w_q \notin B\text{-Free}_{\mathcal{B}}(q)$. As there are only finitely many states in $[q_\sim]_\sim$, it is then possible to build a word $w = w_{q_1} \ldots w_{q_n}$

such that for all $1 \leq i < n$, $\delta^*(q_i, w_{q_i}) = q_{i+1}$, $\delta^*(q_n, w_{q_n}) = q_1$, and for all $1 \leq i \leq n$, $w_{q_i} \in C_{[q_\sim]} \setminus C'_{[q_\sim]}$ and $w_{q_i} \notin B\text{-Free}_\mathcal{B}(q_i)$. Word w^ω is accepted from q_1 as it sees infinitely many Büchi transitions, so it is in $(q_1)^{-1}W = q_\sim^{-1}W$. However, if we consider w^ω as an infinite word on $C_{[q_\sim]}$, then it is not in $W_{[q_\sim]} = \text{Büchi}(C'_{[q_\sim]})$ as every letter of the word is in $C_{[q_\sim]} \setminus C'_{[q_\sim]}$. This yields a contradiction. $\qquad\square$

We use the previous result in a straightforward way to exhibit a state q_{max} whose non-B-free words in $C_{[q_\sim]}$ are exactly the words in $C'_{[q_\sim]}$. The reader may notice that, echoing the proof of the prefix-independent case (Subsection 8.4.1), the state q_{max} given by Corollary 8.4.9 is actually a B-free-maximum among states in $[q_\sim]_\sim$.

> **Corollary 8.4.9** *There exists $q_{max} \in [q_\sim]_\sim$ such that for all $w \in C_{[q_\sim]}$, $w \in C'_{[q_\sim]}$ if and only if $w \notin B\text{-Free}_\mathcal{B}(q_{max})$.*

Proof. By the second item of Lemma 8.4.8, we take $q_{max} \in [q_\sim]_\sim$ such that, for all $w \in C_{[q_\sim]} \setminus C'_{[q_\sim]}$, $w \in B\text{-Free}_\mathcal{B}(q_{max})$. Let $w \in C_{[q_\sim]}$. The property we already have on q_{max} gives us by contrapositive that $w \notin B\text{-Free}_\mathcal{B}(q_{max})$ implies that $w \in C'_{[q_\sim]}$. Reciprocally, the first item of Lemma 8.4.8 gives us that if $w \in C'_{[q_\sim]}$, then $w \notin B\text{-Free}_\mathcal{B}(q_{max})$. $\qquad\square$

From now on, we assume that $q_{max} \in [q_\sim]_\sim$ **is a state having the property of Corollary 8.4.9**. We show that W can be recognized by a smaller DBA consisting of DBA $\mathcal{B}$ in which all the states in $[q_\sim]_\sim$ have been merged into the single state q_{max}, by redirecting all incoming transitions of $[q_\sim]_\sim$ to q_{max}. We assume w.l.o.g. that if $q_{init} \in [q_\sim]_\sim$, then $q_{init} = q_{max}$ (this does not change the objective recognized by $\mathcal{B}$, and will be convenient in the upcoming construction). We consider DBA $\mathcal{B}' = (Q', C, q'_{init}, \delta', B')$ with

- ▸ $Q' = (Q \setminus [q_\sim]_\sim) \cup \{q_{max}\}$, $q'_{init} = q_{init}$,
- ▸ for $q \in Q'$ and $c \in C$, if $\delta(q, c) \in [q_\sim]_\sim$, then $\delta'(q, c) = q_{max}$; otherwise, $\delta'(q, c) = \delta(q, c)$,
- ▸ for $q \in Q'$ and $c \in C$, $(q, c) \in B'$ if and only if $(q, c) \in B$.

We also assume that states that are not reachable from q'_{init} in $\mathcal{B}'$ are removed from Q'.

> **Lemma 8.4.10** *The objective recognized by $\mathcal{B}'$ is also W.*

Proof. Let $w = c_1 c_2 \ldots \in C^\omega$, $\varrho = (q_0, c_1)(q_1, c_2) \ldots$ be the run of $\mathcal{B}$ on w, and $\varrho' = (q'_0, c_1)(q'_1, c_2) \ldots$ be the run of $\mathcal{B}'$ on w. We have that $q_0 = q'_0$, but states in both runs may not coincide after a state in $[q_\sim]_\sim$ has been reached. Yet, we show inductively that

$$\forall i \geq 0, q_i \sim_\mathcal{B} q'_i. \tag{8.3}$$

It is true for $i = 0$ (we even have equality in this case), and if $q_n \sim_\mathcal{B} q'_n$, then by construction of the transitions of $\mathcal{B}'$ and Lemma 2.8.10, we still have $q_{n+1} \sim_\mathcal{B} q'_{n+1}$.

We want to show that w is accepted by $\mathcal{B}$ if and only if it is accepted by $\mathcal{B}'$. This is clear if w never goes through a state in $[q_\sim]_\sim$ (as the same transitions are then taken in $\mathcal{B}$ and $\mathcal{B}'$).

We first assume that run ϱ visits $[q_\sim]_\sim$ finitely many times, and that the last visit to $[q_\sim]_\sim$ happens in q_n for some $n \geq 0$. Notice that run ϱ' also visits $[q_\sim]_\sim$ for the last time in q'_n by Equation (8.3). Therefore, word $c_{n+1}c_{n+2}\ldots$ is accepted from q'_n in $\mathcal{B}'$ if and only if it is accepted from q'_n in $\mathcal{B}$: all the subsequent transitions coincide. As $q_n \sim_\mathcal{B} q'_n$, we have moreover that $c_{n+1}c_{n+2}\ldots$ is accepted from q'_n in $\mathcal{B}'$ if and only if it is accepted from q_n in $\mathcal{B}$. This implies that w is accepted by $\mathcal{B}$ if and only if it is accepted by $\mathcal{B}'$.

We now assume that run ϱ visits $[q_\sim]_\sim$ infinitely many times. We decompose w inductively into a prefix $w_{q_\sim}$ reaching $[q_\sim]_\sim$ followed by cycles $w_1, w_2, \ldots$ on $[q_\sim]_\sim$. Formally, let $w_{q_\sim}$ be any finite prefix of w such that $\delta^*(q_{\text{init}}, w_{q_\sim}) \in [q_\sim]_\sim$. By induction hypothesis, assume $w_{q_\sim}w_1 \ldots w_n$ is a prefix of w such that $\delta^*(q_{\text{init}}, w_{q_\sim}w_1 \ldots w_n) \in [q_\sim]_\sim$. We define $w_{n+1} \in C^+$ as the shortest non-empty word such that $w_{q_\sim}w_1 \ldots w_n w_{n+1}$ is a prefix of w and $\delta^*(q_{\text{init}}, w_{q_\sim}w_1 \ldots w_n w_{n+1}) \in [q_\sim]_\sim$. We have $w = w_{q_\sim}w_1 w_2 \ldots$ by construction. By Equation (8.3), we also have that for all $n \geq 0$, $(\delta')^*(q'_{\text{init}}, w_q w_1 \ldots w_n) \sim_\mathcal{B} q_\sim$, so $(\delta')^*(q'_{\text{init}}, w_q w_1 \ldots w_n) = q_{\max}$ as this is the only state left in that class in $\mathcal{B}'$.

If w is accepted by $\mathcal{B}$, then for infinitely many $i \geq 1$, word w_i is in $C'_{[q_\sim]}$. By Corollary 8.4.9, all these infinitely many words are not in $B\text{-Free}_\mathcal{B}(q_{\max})$ and therefore see a Büchi transition when read from $q_{\max}$, so w is also accepted by $\mathcal{B}'$.

If w is rejected by $\mathcal{B}$, then there exists $n \geq 1$ such that for all $n' \geq n$, $w_{n'} \in C_{[q_\sim]} \setminus C'_{[q_\sim]}$. By Corollary 8.4.9, for all $n' \geq n$, $w_{n'}$ is in $B\text{-Free}_\mathcal{B}(q_{\max})$ and therefore does not see a Büchi transition when read from $q_{\max}$, so w is also rejected by $\mathcal{B}'$. $\qquad\square$

We have all the arguments to show our goal for the section (Proposition 8.4.1), that is, to show that W can be recognized by a Büchi automaton built on top of $\mathcal{S}_W$.

Proof of Proposition 8.4.1. We have shown in Lemma 8.4.10 how to merge an equivalence class for $\sim$ into a single state, while still recognizing the same objective. Repeating this construction for each equivalence class with two or more states, we end up with a DBA with exactly one state per equivalence class for $\sim$ still recognizing W. By definition of $\mathcal{S}_W$, this DBA is necessarily built on top of (some automaton structure isomorphic to) $\mathcal{S}_W$. $\qquad\square$

8.5 Sufficiency of the conditions

We show that a DBA $\mathcal{B}$ with the three conditions from Subsection 8.3.1 (recognizing an $\mathcal{M}_{\text{triv}}$-progress-consistent objective having a total prefix preorder and being recognizable by a Büchi automaton built on top of $\mathcal{S}_W$) recognizes a half-positional objective. As these three conditions have been shown to be necessary for the half-positionality of objectives recognized by a DBA, this will imply a characterization of half-positionality.

Our main technical tool is to construct, thanks to these three conditions, a family of *completely well-monotonic universal graphs*. The existence of such objects implies, thanks to recent results by Ohlmann [Ohl23], that $\mathcal{P}_1$ has memoryless optimal strategies, even in two-player arenas of arbitrary cardinality.

[Ohl23]: Ohlmann (2023), *Characterizing Positionality in Games of Infinite Duration over Infinite Graphs*

8.5.1 Completely well-monotonic universal graphs

We fix extra terminology about graphs only used in Section 8.5, and recall the relevant results from [Ohl23].

An *(edge-colored) graph* $G = (V, E)$ is given by a non-empty set of *vertices* V (of any cardinality) and a set of *edges* $E \subseteq V \times C \times V$. For convenience, we write $v \xrightarrow{c} v'$ if $(v, c, v') \in E$. We assume graphs to be *non-blocking*: for all $v \in V$, there exists $(v', c, v'') \in E$ such that $v = v'$. We allow graphs with infinite branching. An edge-colored graph is roughly the same as an arena, except that we do specify which player controls each vertex. For $v \in V$, an *infinite path of G from v* is an infinite sequence of edges $\rho = (v_0, c_1, v'_1)(v_1, c_2, v'_2) \ldots \in E^{\omega}$ such that $v_0 = v$ and for all $i \geq 1$, $v'_i = v_i$. A *finite path of G from v* is a finite prefix in E^* of an infinite path of G from v.

We would like to define a reasonable notion of morphism for graphs that also deals with information given by an objective W. Let G be a graph and $W \subseteq C^{\omega}$ be an objective. We say that a vertex v of G *satisfies W* if for all infinite paths $v_0 \xrightarrow{c_1} v_1 \xrightarrow{c_2} \ldots$ from v, we have $c_1 c_2 \ldots \in W$. A graph can therefore be thought of as a one-player arena of $\mathcal{P}_2$: a vertex satisfies W if and only if $\mathcal{P}_2$ cannot win from v, even when controlling all the vertices.

Given two graphs $G = (V, E)$ and $G' = (V', E')$, a *(graph) morphism from G to G'* is a function $\phi \colon V \to V'$ such that $(v_1, c, v_2) \in E$ implies $(\phi(v_1), c, \phi(v_2)) \in E'$. A morphism ϕ from G to G' is *W-preserving* if for all $v \in V$, v satisfies W implies that $\phi(v)$ satisfies W. Notice that if $\phi(v)$ satisfies W, then v satisfies W, as any path $v \xrightarrow{c_1} v_1 \xrightarrow{c_2} \ldots$ of G implies the existence of a path $\phi(v) \xrightarrow{c_1} \phi(v_1) \xrightarrow{c_2} \ldots$ of G' — there are "more paths" in G'. For κ a cardinal, a graph $\mathcal{U}$ is *(κ, W)-universal* if for all graphs G of cardinality $< \kappa$, there is a W-preserving morphism from G to $\mathcal{U}$.

We consider a graph $G = (V, E)$ along with a total order $\leq$ on its vertex set V. We say that G is *monotonic* if for all $v, v', v'' \in V$, for all $c \in C$,

▸ $(v \xrightarrow{c} v'$ and $v' \geq v'') \implies v \xrightarrow{c} v''$, and
▸ $(v \geq v'$ and $v' \xrightarrow{c} v'') \implies v \xrightarrow{c} v''$.

This means that (i) whenever there is an edge $v \xrightarrow{c} v'$, there is also an edge with color c from v to all vertices smaller than v' for $\leq$, and (ii) whenever $v \geq v'$, then v has at least the same outgoing edges as v'. Graph G is *well-monotonic* if it is monotonic and the total order $\leq$ is well-founded (i.e., any set of vertices has a minimum). Graph G is *completely well-monotonic* if it is well-monotonic and there exists a vertex $\top \in V$ maximum for $\leq$ such that for all $v \in V, c \in C, \top \xrightarrow{c} v$.

> **Example 8.5.1** We provide an example illustrating these notions with $C = \{a, b\}$ and $W = \mathsf{Büchi}(a)$. This is a special case of our upcoming construction, and it is already discussed in more depth in [Ohl21, Chapter 2]. For θ an ordinal, we define a graph $\mathcal{U}_\theta$ with vertices $U_\theta = \theta \cup \{\top\}$, and such that for all ordinals $\lambda, \lambda' < \theta$,
>
> ▸ $\lambda \xrightarrow{a} \lambda'$, and
> ▸ $\lambda \xrightarrow{b} \lambda'$ if and only if $\lambda' < \lambda$.
>
> Moreover, for all $v \in U_\theta$, we define edges $\top \xrightarrow{a} v$ and $\top \xrightarrow{b} v$. We order vertices using the natural order on the ordinals, and with $\lambda < \top$ for all $\lambda < \theta$.
> Vertex $\top$ does not satisfy W, as there is an infinite path $\top \xrightarrow{b} \top \xrightarrow{b} \ldots$, and $b^\omega \notin W$. All other vertices satisfy W by construction: they cannot reach $\top$, and as reading b decreases the current ordinal, a path cannot have an infinite suffix using only color b (there is no infinite decreasing sequence of ordinals).
> This graph is (κ, W)-universal for $\kappa = |\theta|$. Intuitively, for any graph G with less than κ vertices, a W-preserving morphism from G to $\mathcal{U}_\theta$ can be defined by mapping vertices not satisfying W to $\top$, and vertices satisfying W to an ordinal λ that depends on how long it may take to guarantee seeing an a from them.
> Graph $\mathcal{U}_\theta$ is monotonic, which can be quickly checked with the definition. As $\leq$ is well-founded and there is a vertex $\top$ with the right properties, $\mathcal{U}_\theta$ is even completely well-monotonic. The properties of $\mathcal{U}_\theta$ imply half-positionality of $\mathsf{Büchi}(a)$, thanks to the following theorem.

[Ohl21]: Ohlmann (2021), *Monotonic graphs for parity and mean-payoff games*

We state an important result linking half-positionality and completely well-monotonic universal graphs from [Ohl23].

[Ohl23]: Ohlmann (2023), *Characterizing Positionality in Games of Infinite Duration over Infinite Graphs*

> **Theorem 8.5.2** ([Ohl23]) *Let $W \subseteq C^\omega$ be an objective. If for all cardinals κ, there exists a completely well-monotonic (κ, W)-universal graph, then W is half-positional (over all arenas).*

A more precise result in [Ohl21, Theorem 1.1] can actually be instantiated on more subtle classes of arenas. However, we intend to prove here half-positionality of a family of objectives over *all* arenas, so the above result turns out to be sufficient.

[Ohl21]: Ohlmann (2021), *Monotonic graphs for parity and mean-payoff games*

Remark 8.5.3 One of the strengths of [Ohl23] is that the converse of Theorem 8.5.2 holds under the existence of a *neutral color*, i.e., a color that can be inserted infinitely often in every infinite word without changing its winning or losing character. A neutral color can always be artificially added to an objective. The question of whether adding a neutral color can make a half-positional objective non-half-positional is open.
In some works (for instance the seminal [DJW97; Zie98]), it is assumed that edges without colors are allowed in arenas, with the restriction that no cycle of the arenas is completely without color. This is roughly similar to adding a neutral color to the set of colors. In general, allowing for arenas with such edges has an effect on strategy complexity [Cas22]. The existence of a family of completely well-monotonic (κ, W)-universal graphs implies that a neutral color can be added without losing half-positionality of W [Ohl23]. Hence, given our proof technique for half-positional DBA-recognizable objectives, a neutral color (or edges without colors) can always be added for free.

[Ohl23]: Ohlmann (2023), *Characterizing Positionality in Games of Infinite Duration over Infinite Graphs*
Neutral colors were also discussed in Remark 5.5.7.

[DJW97]: Dziembowski et al. (1997), *How Much Memory is Needed to Win Infinite Games?*
[Zie98]: Zielonka (1998), *Infinite Games on Finitely Coloured Graphs with Applications to Automata on Infinite Trees*

[Cas22]: Casares (2022), *On the Minimisation of Transition-Based Rabin Automata and the Chromatic Memory Requirements of Muller Conditions*

8.5.2 Universal graphs for Büchi automata

We show that for a DBA-recognizable objective, the three conditions that were shown to be necessary for half-positionality in Section 8.4 are actually sufficient.

Proposition 8.5.4 (Sufficiency of the three conditions) *Let $W \subseteq C^\omega$ be an objective that has a total prefix preorder, is $\mathcal{M}_{\mathsf{triv}}$-progress-consistent, and is recognizable by a Büchi automaton built on top of $\mathcal{S}_W$. Then, W is half-positional.*

The rest of the section is devoted to the proof of this result, using Theorem 8.5.2. Let $W \subseteq C^\omega$ be an objective with a **total prefix preorder**, that is $\mathcal{M}_{\mathsf{triv}}$**-progress-consistent**, and that is **recognized by a DBA** $\mathcal{B} = (Q, C, q_{\mathsf{init}}, \delta, B)$ **built on top of** $\mathcal{S}_W$ for the rest of this section. We assume as in previous sections that $\mathcal{B}$ **is saturated** (in particular, by Lemma 8.2.4, any B-free path can be extended to a B-free cycle). An implication of the fact that $\mathcal{B}$ is built on top of $\mathcal{S}_W$ that we will use numerous times in the upcoming arguments is that for $q, q' \in Q$, $q \sim q'$ if and only if $q = q'$.

For θ an ordinal, we build a graph $\mathcal{U}_{\mathcal{B}, \theta}$ in the following way.

- We set the vertices as $U_{\mathcal{B}, \theta} = \{(q, \lambda) \mid q \in Q, \lambda < \theta\} \cup \{\top\}$.
- For every transition $\delta(q, c) = q'$ of $\mathcal{B}$,

- if $(q, c) \in B$, then for all ordinals λ, λ', we define an edge $(q, \lambda) \xrightarrow{c} (q', \lambda')$;
- if $(q, c) \notin B$, then for all ordinals λ, λ' s.t. $\lambda' < \lambda$, we define an edge $(q, \lambda) \xrightarrow{c} (q', \lambda')$.
- for $q'' \prec q'$, then for all ordinals λ, λ'', we define an edge $(q, \lambda) \xrightarrow{c} (q'', \lambda'')$.

▶ For all $v \in U_{\mathcal{B}, \theta}$ and $c \in C$, we define an edge $\top \xrightarrow{c} v$.

We order the vertices lexicographically: $(q, \lambda) \leq (q', \lambda')$ if $q \prec q'$ or ($q = q'$ and $\lambda \leq \lambda'$), and we define $\top$ as the maximum for $\leq$ (i.e., $(q, \lambda) < \top$ for all $q \in Q$, $\lambda < \theta$).

Graph $\mathcal{U}_{\mathcal{B}, \theta}$ is built such that on the one hand, it is sufficiently large and has sufficiently many edges so that there is a morphism from any graph G (of cardinality smaller than some function of $|\theta|$) to $\mathcal{U}_{\mathcal{B}, \theta}$. On the other hand, for the morphism to be W-preserving, at least some vertices of $\mathcal{U}_{\mathcal{B}, \theta}$ need to satisfy W, which imposes a restriction on the infinite paths from vertices. Graph $\mathcal{U}_{\mathcal{B}, \theta}$ is actually built so that for any automaton state $q \in Q$ and ordinal $\lambda < \theta$, the vertex (q, λ) satisfies $q^{-1}W$ (see Lemma 8.5.9). The intuitive idea is that for a non-Büchi transition $(q, c) \notin B$ of the automaton such that $\delta(q, c) = q'$, a c-colored edge from a vertex (q, λ) in the graph either (i) reaches a vertex with first component q', in which case the ordinal must decrease on the second component, or (ii) reaches a vertex with first component $q'' \prec q'$, with no restriction on the second component, but therefore with fewer winning continuations. Using $\mathcal{M}_{\text{triv}}$-progress-consistency and the fact that there is no infinitely decreasing sequence of ordinals, we can show that this implies that no infinite path in $\mathcal{U}_{\mathcal{B}, \theta}$ corresponds to an infinite run in the automaton visiting only non-Büchi transitions.

We state two properties that directly follow from the definition of $\mathcal{U}_{\mathcal{B}, \theta}$:

$$\text{if } (q, \lambda) \xrightarrow{c} (q', \lambda'), \text{ then } q' \preceq \delta(q, c); \tag{8.4}$$

$$\text{if } (q, \lambda) \xrightarrow{c} (q', \lambda') \text{ and } \lambda'' \leq \lambda', \text{ then } (q, \lambda) \xrightarrow{c} (q', \lambda''). \tag{8.5}$$

Example 8.5.5 We consider again the DBA $\mathcal{B}$ from Example 8.3.3, recognizing the words seeing a infinitely many times, or a twice in a row at some point. We represent the graph $\mathcal{U}_{\mathcal{B}, \theta}$, with $\theta = \omega$ in Figure 8.10.

In order to use Theorem 8.5.2, we show that the graph $\mathcal{U}_{\mathcal{B}, \theta}$ is completely well-monotonic (Lemma 8.5.6) and, for all cardinals κ, is (κ, W)-universal for sufficiently large θ (Proposition 8.5.10).

Lemma 8.5.6 *Graph $\mathcal{U}_{\mathcal{B}, \theta}$ is completely well-monotonic.*

Proof. The order $\leq$ on the vertices is well-founded, and there exists a vertex $\top \in U_{\mathcal{B}, \theta}$ maximum for $\leq$ such that for all $v \in U_{\mathcal{B}, \theta}$, $c \in C$, $\top \xrightarrow{c} v$.
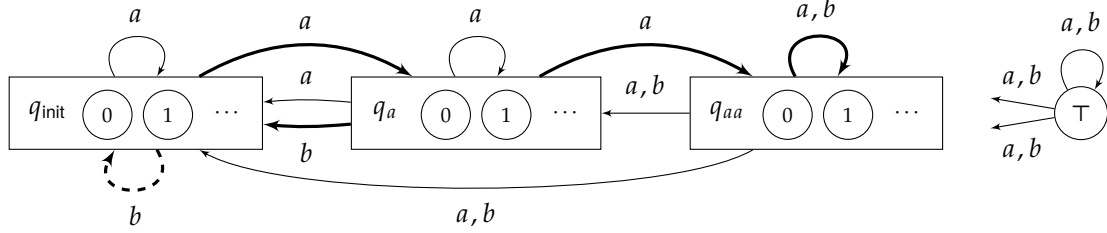
Figure 8.10: The graph $\mathcal{U}_{\mathcal{B},\omega}$, where $\mathcal{B}$ is the automaton from Example 8.3.3 ($\mathcal{L}(\mathcal{B}) = \text{Büchi}(a) \cup C^* aa C^\omega$). The dashed edge with color b indicates that $(q_{\text{init}}, \lambda) \xrightarrow{b} (q_{\text{init}}, \lambda')$ if and only if $\lambda' < \lambda$ (it corresponds to the only non-Büchi transition in $\mathcal{B}$). Elsewhere, an edge between two rectangles labeled q, q' with color c means that for all ordinals λ, λ', $(q, \lambda) \xrightarrow{c} (q', \lambda')$. Thick edges correspond to the original transitions of $\mathcal{B}$. There are edges from $\top$ to all vertices of the graph with colors a and b. Vertices are totally ordered from left to right.

To show that $\mathcal{U}_{\mathcal{B},\theta}$ is completely well-monotonic, it now suffices to show that $\mathcal{U}_{\mathcal{B},\theta}$ is monotonic.

The first item of the monotonicity definition follows from the construction of the graph. We assume that $(q, \lambda) \xrightarrow{c} (q', \lambda')$ and $(q', \lambda') \geq (q'', \lambda'')$, and we show that $(q, \lambda) \xrightarrow{c} (q'', \lambda'')$. By Equation (8.4), we have $q' \leq \delta(q, c)$. The inequality $(q', \lambda') \geq (q'', \lambda'')$ means by definition that $q'' < q'$ or $(q'' = q'$ and $\lambda'' \leq \lambda')$. If $q'' < q'$, we obtain that $q'' < \delta(q, c)$, so we also have $(q, \lambda) \xrightarrow{c} (q'', \lambda'')$. If $q'' = q'$ and $\lambda'' \leq \lambda'$, then we also have $(q, \lambda) \xrightarrow{c} (q', \lambda'')$ by Equation (8.5), which is what we want as $q' = q''$.

The second item of the monotonicity definition is slightly more involved and follows from $\mathcal{M}_{\text{triv}}$-progress-consistency, the fact that the prefix preorder is total, and the saturation of $\mathcal{B}$. We assume that $(q, \lambda) \geq (q', \lambda')$ and $(q', \lambda') \xrightarrow{c} (q'', \lambda'')$, and we show that $(q, \lambda) \xrightarrow{c} (q'', \lambda'')$. The assumption $(q, \lambda) \geq (q', \lambda')$ implies that $q \geq q'$, so $\delta(q, c) \geq \delta(q', c)$ by Lemma 2.8.10. Moreover, $(q', \lambda') \xrightarrow{c} (q'', \lambda'')$ implies that $\delta(q', c) \geq q''$ by Equation (8.4). Hence, $\delta(q, c) \geq q''$. If $\delta(q, c) > q''$, then $(q, \lambda) \xrightarrow{c} (q'', \lambda'')$ by definition of the graph. The same holds if $\delta(q, c) = q''$ and $(q, c) \in B$.

It is left to discuss the case $\delta(q, c) = q''$ and $(q, c) \notin B$. By the above inequalities, this implies that we also have $\delta(q', c) = q''$.

- If $q = q'$, then $\lambda \geq \lambda'$. Moreover, as $(q', c) = (q, c) \notin B$, the existence of edge $(q', \lambda') \xrightarrow{c} (q'', \lambda'')$ implies that $\lambda' > \lambda''$. So $\lambda > \lambda''$ and we also have $(q, \lambda) \xrightarrow{c} (q'', \lambda'')$.
- We show that $q' < q$ is not possible — we assume it holds and draw a contradiction. As $(q, c) \notin B$, we have $c \in B\text{-Free}_{\mathcal{B}}(q)$. By Lemma 8.2.4, there is $w \in C^*$ such that $cw \in B\text{-FreeCycles}_{\mathcal{B}}(q)$. As $\delta(q, c) = \delta(q', c)$ and $\delta^*(q, cw) = q$, we have $\delta^*(q', cw) = q$. As $q' < q$, by $\mathcal{M}_{\text{triv}}$-progress-consistency, the word $(cw)^\omega$ must be accepted by $\mathcal{B}$ when read from q'. It must therefore also be accepted when read from q (as $q' < q$), which contradicts that $cw \in B\text{-FreeCycles}_{\mathcal{B}}(q)$. $\qquad\square$

We now intend to show (κ, W)-universality of some $\mathcal{U}_{\mathcal{B},\theta}$ for each cardinal κ. The next two Lemmas 8.5.7 and 8.5.8 give insight into properties of the paths of $\mathcal{U}_{\mathcal{B},\theta}$, to then establish which vertices of $\mathcal{U}_{\mathcal{B},\theta}$ satisfy W (Lemma 8.5.9). Understanding which vertices satisfy W is helpful to later define a W-preserving morphism into $\mathcal{U}_{\mathcal{B},\theta}$. We first show that paths in this graph "underapproximate" corresponding runs in the automaton: a finite path $\gamma = (q_0, \lambda_0) \xrightarrow{c_1} \ldots \xrightarrow{c_n} (q_n, \lambda_n)$ in $\mathcal{U}_{\mathcal{B},\theta}$ visits vertices labeled with automaton states at most as large (for $\preceq$) as the corresponding states visited by the finite run from q_0 on $c_1 \ldots c_n$.

> **Lemma 8.5.7** Let $\gamma = (q_0, \lambda_0) \xrightarrow{c_1} \ldots \xrightarrow{c_n} (q_n, \lambda_n)$ be a finite path of $\mathcal{U}_{\mathcal{B},\theta}$ and $w = c_1 \ldots c_n$. Then, $q_n \preceq \delta^*(q_0, w)$.

Proof. We proceed by induction on the length n of γ. If $n = 0$, then $w = \varepsilon$, so $q_n = q_0 = \delta^*(q_0, w)$, and the result holds. If $n \geq 1$, we assume by induction hypothesis that $q_{n-1} \preceq \delta^*(q_0, c_1 \ldots c_{n-1})$. By Lemma 2.8.10, we have $\delta(q_{n-1}, c_n) \preceq \delta(\delta^*(q_0, c_1 \ldots c_{n-1}), c_n) = \delta^*(q_0, w)$. By Equation (8.4), if $(q_{n-1}, \lambda_{n-1}) \xrightarrow{c_n} (q_n, \lambda_n)$, then $q_n \preceq \delta(q_{n-1}, c_n)$. By transitivity, $q_n \preceq \delta^*(q_0, w)$. $\qquad\square$

We now show that in $\mathcal{U}_{\mathcal{B},\theta}$, a finite path that goes back to its initial value w.r.t. the first component without decreasing the ordinal necessarily induces an accepted word when repeated.

> **Lemma 8.5.8** Let $\gamma = (q_0, \lambda_0) \xrightarrow{c_1} \ldots \xrightarrow{c_n} (q_n, \lambda_n)$ be a finite path of $\mathcal{U}_{\mathcal{B},\theta}$ with $n \geq 1$, $q_0 = q_n$, and $\lambda_0 \leq \lambda_n$. Let $w = c_1 \ldots c_n$. Then, $w^\omega \in q_0^{-1} W$.

Proof. Let $\mathcal{B}(q_0, w) = (q_0', c_1) \ldots (q_{n-1}', c_n)$ be the finite run of $\mathcal{B}$ obtained by reading w from q_0. States $q_0, \ldots, q_n$ correspond to the first component of the vertices visited by γ in $\mathcal{U}_{\mathcal{B},\theta}$, whereas $q_0', \ldots, q_{n-1}', q_n' = \delta(q_{n-1}', c_n)$ are the states visited by the finite word w in $\mathcal{B}$. We have that $q_0 = q_0'$, but the subsequent states may or may not correspond. By Lemma 8.5.7, we still know that for all i, $0 \leq i \leq n$, $q_i \preceq q_i'$. In particular, $q_0 = q_n \preceq \delta^*(q_0, w) = q_n'$. We distinguish three cases, depending on whether $q_0 \prec \delta^*(q_0, w)$ or $q_0 \sim \delta^*(q_0, w)$ (which implies $q_0 = \delta^*(q_0, w)$ as $\mathcal{B}$ is built on top of its prefix classifier), and depending on whether $q_i = q_i'$ for all $0 \leq i \leq n$ or not.

- ▶ If $q_0 \prec \delta^*(q_0, w)$, by $\mathcal{M}_{\text{triv}}$-progress-consistency, $w^\omega \in q_0^{-1} W$.
- ▶ If $q_0 = \delta^*(q_0, w)$ and for all i, $0 \leq i \leq n$, $q_i = q_i'$ (i.e., γ only uses edges that directly correspond to transitions of the automaton $\mathcal{B}$), then for the ordinal to be greater than or equal to its starting value, some Büchi transition has to be taken since non-Büchi transitions strictly decrease the ordinal on the second component. Hence, w is not a B-free cycle from q_0, so $w^\omega \in q_0^{-1} W$.
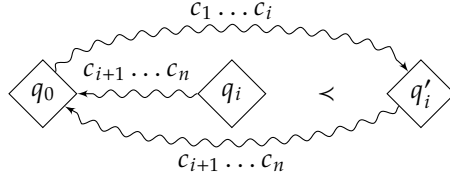
Figure 8.11: Situation in the proof of Lemma 8.5.8.

▶ If $q_0 = \delta^*(q_0, w)$ and for some index i, $1 \leq i < n$, we have $q_i \prec q_i'$ (i.e., γ takes at least one edge that does not correspond to a transition of the automaton). We represent the situation in Figure 8.11. We know that $\delta^*(q_i', c_{i+1} \ldots c_n) = q_n' \sim q_0$. By Lemma 2.8.10, as $q_i \prec q_i'$, this implies that $\delta^*(q_i, c_{i+1} \ldots c_n) \leq q_0$. Also, in the graph, there is a path from q_i to q_0 with colors $c_{i+1} \ldots c_n$. Therefore, by Lemma 8.5.7, $q_0 \leq \delta^*(q_i, c_{i+1} \ldots c_n)$. Thus, $q_0 \sim \delta^*(q_i, c_{i+1} \ldots c_n)$, and as $\mathcal{B}$ is built on top of $\mathcal{S}_W$, $q_0 = \delta^*(q_i, c_{i+1} \ldots c_n)$. Therefore, $q_i \prec \delta^*(q_i, c_{i+1} \ldots c_n c_1 \ldots c_i) = q_i'$. By $\mathcal{M}_{\text{triv}}$-progress-consistency, $(c_{i+1} \ldots c_n c_1 \ldots c_i)^\omega \in q_i^{-1} W$. As $\delta^*(q_i, c_{i+1} \ldots c_n) = q_0$, this implies that $(c_1 \ldots c_n)^\omega = w^\omega \in q_0^{-1} W$. □

We show that every vertex (q_0, λ_0) such that $q_0 \leq q_{\text{init}}$ satisfies W.

> **Lemma 8.5.9** *Let $q_0 \in Q$. For all ordinals $\lambda_0 < \theta$, the vertex (q_0, λ_0) of $\mathcal{U}_{\mathcal{B}, \theta}$ satisfies $q_0^{-1} W$. In particular, if $q_0 \leq q_{\text{init}}$, (q_0, λ_0) satisfies W.*

Proof. Let $\rho = (q_0, \lambda_0) \xrightarrow{c_1} (q_1, \lambda_1) \xrightarrow{c_2} \ldots$ be an infinite path of $\mathcal{U}_{\mathcal{B}, \theta}$ from (q_0, λ_0) and $w = c_1 c_2 \ldots$ be the sequence of colors along its edges. We show that $w \in q_0^{-1} W$.

Let $q_i' = \delta^*(q_0, c_1 \ldots c_i)$ be the state of $\mathcal{B}$ reached after reading the first i colors of w. We claim that there are two states $q, q' \in Q$ occurring infinitely often in the sequences $(q_i)_{i \geq 0}$, $(q_i')_{i \geq 0}$, respectively, and an increasing sequence of indices $(i_k)_{k \geq 1}$ satisfying that for all $k \geq 1$,

▶ $q_{i_k} = q$, $q_{i_k}' = q'$, and
▶ $\lambda_{i_k} \leq \lambda_{i_{k+1}}$.

Indeed, we can first choose a state q appearing infinitely often in $(q_i)_{i \geq 0}$ and pick a sequence $(i_j)_{j \geq 1}$ such that $q_{i_j} = q$ and $(\lambda_{i_j})_{j \geq 1}$ is not decreasing (this is possible since there is no infinite decreasing sequence of ordinals). Then, we can just pick q' appearing infinitely often in $(q_{i_j}')_{j \geq 1}$ and extract the subsequence corresponding to its occurrences.

Let $q, q' \in Q$, $(i_k)_{k \geq 1}$ satisfying the above properties. Let $w_0 = c_1 \ldots c_{i_1}$ and for $k \geq 1$, let $w_k = c_{i_k + 1} \ldots c_{i_{k+1}} \in C^+$ be the colors over the edges in ρ from (q_{i_k}, λ_{i_k}) to $(q_{i_{k+1}}, \lambda_{i_{k+1}})$.

By Lemma 8.5.7, it holds that $q \leq q' = \delta^*(q_0, w_0)$. Moreover, since $\lambda_{i_k} \leq \lambda_{i_{k+1}}$, by Lemma 8.5.8 we have that $w_k^\omega \in q^{-1} W \subseteq (q')^{-1} W$ for all $k \geq 1$. We conclude that for all $k \geq 1$, the word w_k labels a cycle over q' in $\mathcal{B}$ visiting some Büchi transition, and therefore $w = w_0 w_1 w_2 \ldots \in q_0^{-1} W$. □

We now have all the tools to show, for all cardinals κ, (κ, W)-universality of $\mathcal{U}_{\mathcal{B},\theta}$ for sufficiently large θ.

> **Proposition 8.5.10** *Let κ be a cardinal, and θ' be an ordinal such that $\kappa \leq |\theta'|$. Let $\theta = |Q| \cdot \theta'$. Graph $\mathcal{U}_{\mathcal{B},\theta}$ is (κ, W)-universal.*

Proof. Let $G = (V, E)$ be a graph such that $|V| < \kappa$ (in particular, $|V| < |\theta'|$). For $v \in V$, let $q_v \in Q \cup \{\top\}$ be the smallest automaton state (for $\leq$) such that v satisfies $q_v^{-1}W$, or $\top$ if it satisfies none of them. We remark that $q_v \leq q$ if and only if v satisfies $q^{-1}W$. To show that there is a W-preserving morphism from G to $\mathcal{U}_{\mathcal{B},\theta}$, we follow the six steps outlined in [Ohl23, Lemma 4.3].

<div style="float:right; width:30%; font-size:small;">

It may seem surprising that we use $|Q| \cdot \theta'$ vertices for each automaton state and not just θ' vertices. We will illustrate in Remark 8.5.12 that this is necessary for our proof technique.

[Ohl23]: Ohlmann (2023), *Characterizing Positionality in Games of Infinite Duration over Infinite Graphs*

We illustrate this induction on a concrete case in Example 8.5.11, which can be followed along the proof.

</div>

(i) In this first step, we classify and order vertices of G in an inductive way, which will later be used to map them to vertices of $\mathcal{U}_{\mathcal{B},\theta}$. For $q \in Q$ and λ an ordinal, we define by transfinite induction

$$V_\lambda^q = \{v \in V \mid q_v \leq q, \text{ and}$$

$$\forall c \in C, \left(v \xrightarrow{c} v' \implies ((q, c) \in B \text{ or } \exists \eta < \lambda, v' \in V_\eta^{\delta(q,c)})\right)\}.$$

Intuitively, for v to be in V_λ^q, it has to satisfy $q^{-1}W$ and to guarantee that a Büchi transition is seen "soon" when colors of paths from v are read from q in $\mathcal{B}$ (how soon depends on the value of λ). We remark that for each state $q \in Q$, the sequence $(V_\lambda^q)_\lambda$ is non-decreasing: for $\lambda \leq \lambda'$, $V_\lambda^q \subseteq V_{\lambda'}^q$. The subsequent steps mostly follow from this definition.

(ii) Let $V^q = \bigcup_\lambda V_\lambda^q$. We show that if v satisfies $q^{-1}W$, then it is in V^q. Assume that $v \notin V^q$. If $q < q_v$, then we immediately have that v does not satisfy $q^{-1}W$. If $q_v \leq q$, then v has an outgoing edge $v \xrightarrow{c} v'$ such that $(q, c) \notin B$ and $v' \notin \bigcup_\lambda V_\lambda^{\delta(q,c)}$. By induction, we build an infinite path from v whose projection in $\mathcal{B}$ only sees non-Büchi transitions, so v does not satisfy $q^{-1}W$.

(iii) In this step and the next one, we show that there is no use in considering ordinals beyond θ in our construction. We first show that if for all $q \in Q$, $V_\lambda^q = V_{\lambda+1}^q$, then for all $q \in Q$ and all $\lambda' \geq \lambda$, $V_\lambda^q = V_{\lambda'}^q$. For $\lambda \leq \lambda'$, we always have $V_\lambda^q \subseteq V_{\lambda'}^q$. For the other inclusion, we assume by transfinite induction that $V_\lambda^{q'} = V_\eta^{q'}$ for all $q' \in Q$ and for all η such that $\lambda \leq \eta < \lambda'$. Let $v \in V_{\lambda'}^q$. Every edge $v \xrightarrow{c} v'$ either satisfies $(q, c) \in B$, or there exists $\eta < \lambda'$ such that $v' \in V_\eta^{\delta(q,c)}$. Since $V_\eta^{\delta(q,c)} \subseteq V_\lambda^{\delta(q,c)}$ by induction hypothesis, we have $v' \in V_\lambda^{\delta(q,c)}$. Hence, $v \in V_{\lambda+1}^q = V_\lambda^q$.

(iv) We prove that there exists $\lambda < \theta$ such that for all $q \in Q$, $V_\lambda^q = V_{\lambda+1}^q$. If not, using the axiom of choice, we can build a map $\psi \colon \theta \to Q \times V$ such that for $\lambda < \theta$, $\psi(\lambda) = (q, v)$ for some $q \in Q$ and $v \in V_{\lambda+1}^q \setminus V_\lambda^q$. This map is injective, as any pair (q, v) can be chosen at most once (as

$(V_\lambda^q)_\lambda$ is non-decreasing). This implies that $|\theta| = |Q| \cdot |\theta'| \leq |Q| \cdot |V|$, a contradiction since $|V| < |\theta'|$.

Using additionally Item (iii), we deduce that there exists $\lambda < \theta$ such that for all $\lambda' \geq \lambda$, $V_\lambda^q = V_{\lambda'}^q$.

(v) Let $\phi \colon V \to U_{\mathcal{B},\theta}$ be such that

$$\phi(v) = \begin{cases} (q_v, \min\{\lambda \mid v \in V_\lambda^{q_v}\}) & \text{if } q_v \prec \top, \\ \top & \text{if } q_v = \top. \end{cases}$$

By Item (ii), for all $v \in V$, there exists λ such that $v \in V_\lambda^{q_v}$, so $\{\lambda \mid v \in V_\lambda^{q_v}\}$ is non-empty. By Item (iv), we have that if $\phi(v) = (q, \lambda)$, then $\lambda < \theta$, so the image of ϕ is indeed in $U_{\mathcal{B},\theta}$. We show that ϕ is W-preserving: if v satisfies W, then $q_v \preceq q_{\text{init}}$, so by Lemma 8.5.9, $\phi(v)$ also satisfies W.

(vi) We show that ϕ is a graph morphism. Let $v \xrightarrow{c} v'$ be an edge of G — we need to show that $\phi(v) \xrightarrow{c} \phi(v')$ is an edge of $U_{\mathcal{B},\theta}$. If $\phi(v) = \top$, this is clear as there are all possible outgoing edges from $\top$. If not, we denote $\phi(v) = (q, \lambda)$ and $\phi(v') = (q', \lambda')$. We have that v satisfies $q^{-1}W$. Thus, v' must satisfy $\delta(q, c)^{-1}W$. This implies that $q' \preceq \delta(q, c)$. We distinguish two cases.

- ▶ If $(q, c) \in B$, then by construction of $U_{\mathcal{B},\theta}$, there are c-colored edges from (q, λ) to $(\delta(q, c), \lambda'')$ for all ordinals λ''. By monotonicity, as $q' \preceq \delta(q, c)$, there is also a c-colored edge from (q, λ) to (q', λ').

- ▶ We assume that $(q, c) \notin B$. If $q' = \delta(q, c)$, this means that there exists $\eta < \lambda$ such that $v' \in V_\eta^{q'}$. Therefore, $\lambda' \leq \eta < \lambda$. So the edge $(q, \lambda) \xrightarrow{c} (q', \lambda')$ exists by construction of $U_{\mathcal{B},\theta}$. If $q' \prec \delta(q, c)$, then by construction of $U_{\mathcal{B},\theta}$, there are c-colored edges from (q, λ) to (q', λ'') for all ordinals λ''.

We have shown in Item (v) and Item (vi) that ϕ is a W-preserving morphism from G to $U_{\mathcal{B},\theta}$. $\qquad\square$

Example 8.5.11 We consider the objective $W = \mathsf{Büchi}(a) \cup C^* aaC^\omega$ recognized by the DBA $\mathcal{B}$ from Example 8.3.3, for which graph $U_{\mathcal{B},\omega}$ was shown in Example 8.5.5. We redraw this DBA in Figure 8.12 (left) and we discuss how our construction maps the vertices of graph G in Figure 8.12 (right). Notice that v_1, v_2, and v_3 satisfy W, but not v_4 and v_5; more precisely, using notations from the proof of Proposition 8.5.10, we have $q_{v_1} = q_{v_2} = q_{v_3} = q_{\text{init}}$, $q_{v_4} = q_a$, and $q_{v_5} = q_{aa}$.

We build explicitly the sets V_λ^q from Item (i) of the proof of Proposition 8.5.10. All five vertices are in $V_\lambda^{q_{aa}}$ for every ordinal λ, as all vertices satisfy $q_{aa}^{-1}W = C^\omega$, and any transition from q_{aa} is a Büchi transition. For the same reasons, v_1, v_2, v_3, and v_4 are in $V_\lambda^{q_a}$ for all λ (v_5 is not because it does not satisfy objective $q_a^{-1}W$). We determine the sets $V_\lambda^{q_{\text{init}}}$ using the inductive definition. First, v_3 is in $V_\lambda^{q_{\text{init}}}$ for all λ, since $(q_{\text{init}}, a) \in B$.
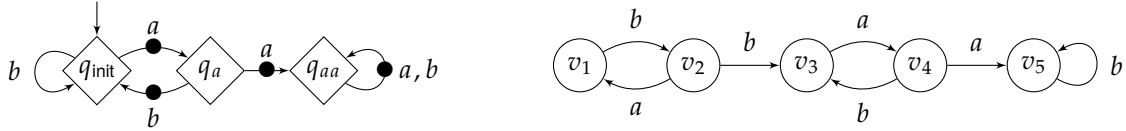
Figure 8.12: DBA recognizing $W = \text{Büchi}(a) \cup C^* aa C^\omega$ (left, redrawn from Example 8.3.3); graph G (right) used in Example 8.5.11, illustrating the proof of Proposition 8.5.10.

Therefore, $v_2 \in V_\lambda^{q_{\text{init}}}$ for $\lambda \geq 1$ (the b-colored edge from v_2 leads to v_3, and $\delta(q_{\text{init}}, b) = q_{\text{init}}$). Finally, $v_1 \in V_\lambda^{q_{\text{init}}}$ for $\lambda \geq 2$ for the same reason. The morphism ϕ from Item (v) assigns $\phi(v_1) = (q_{\text{init}}, 2)$, $\phi(v_2) = (q_{\text{init}}, 1)$, $\phi(v_3) = (q_{\text{init}}, 0)$, $\phi(v_4) = (q_a, 0)$, $\phi(v_5) = (q_{aa}, 0)$.

Remark 8.5.12 We illustrate why, in the statement of Proposition 8.5.10, we use $|Q| \cdot \theta'$ vertices for each automaton state and not just θ'.

Let $C = \{a, b, c\}$, W be the objective recognized by the DBA in Figure 8.13 (left), and G be the graph with two vertices in Figure 8.13 (right). This objective has a total prefix preorder, is $\mathcal{M}_{\text{triv}}$-progress-consistent, is recognized by a DBA built on top of its prefix classifier, and the DBA in Figure 8.13 is saturated. Both vertices of G satisfy $W = q_1^{-1} W$ but do not satisfy $q_0^{-1} W$, so $q_{v_1} = q_{v_2} = q_1$. We have $v_1 \in V_0^{q_3}$ (the base case of the induction), and this inductively implies that $v_2 \in V_1^{q_3}$, $v_1 \in V_2^{q_2}$, $v_2 \in V_3^{q_2}$, $v_1 \in V_4^{q_1}$, $v_2 \in V_5^{q_1}$. We represent these steps in the figure, below the graph. Thus, $\phi(v_1) = (q_1, 4)$ and $\phi(v_2) = (q_1, 5)$. We see here that two copies of each automaton state in our universal graph construction would not have sufficed; we actually used six different indices to fully understand the situation, which is due to the interlacing between the graph and the structure of the automaton.
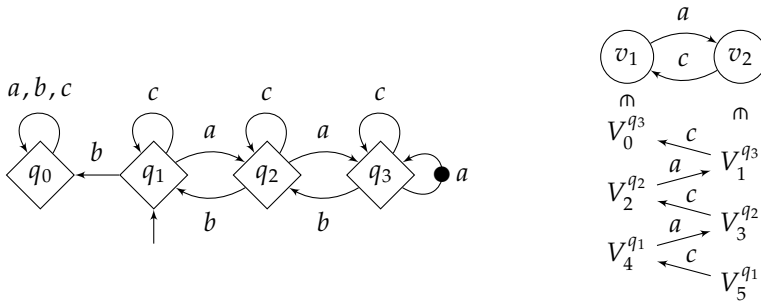
Figure 8.13: DBA (left) and graph G (right) used in Remark 8.5.12 to illustrate the need for $|Q| \cdot \theta'$ vertices for each automaton state in the universal graph used in Proposition 8.5.10.

We conclude this section by proving Proposition 8.5.4, showing that W is half-positional under the three conditions from Theorem 8.3.7.

Proof of Proposition 8.5.4. Using Lemma 8.5.6 and Proposition 8.5.10, we have for all cardinals κ that there exists a completely well-monotonic (κ, W)-universal graph. By Theorem 8.5.2, this implies that W is half-positional. □

8.6 Wrap-up

We have characterized half-positionality for DBA-recognizable objectives. While half-positionality of ω-regular objectives is still not completely understood, this is a novel step in this direction.

Another interesting extension would be to characterize the memory requirements of DBA-recognizable objectives. An intermediate and already seemingly difficult step would be to characterize the memory requirements of objectives recognized by deterministic *weak* automata [Wag79; Sta83; Löd01], already generalizing the characterization for regular objectives (Chapter 7).

[Wag79]: Wagner (1979), *On ω-Regular Sets*
[Sta83]: Staiger (1983), *Finite-State ω-Languages*
[Löd01]: Löding (2001), *Efficient minimization of deterministic weak omega-automata*

CONCLUDING REMARKS

Summary and future prospects 9

We give an overview of the contributions of this thesis. After a brief high-level sketch of our contributions (Section 9.1), we offer two points of view to revisit and appreciate our results: a summary through the various properties defined in this thesis (Subsection 9.1.1), and a summary through the ubiquitous *one-to-two-player lifts* (Subsection 9.1.2). We then describe some promising research directions related to the topic at hand, which we leave for future work (Section 9.2).

9.1 Summary

We studied strategies in two-player zero-sum turn-based games on graphs, focusing on the following question: how complex must the implementation of a strategy be, and how simple can it be in order to make optimal decisions in a game? Our study started from *objectives* (descriptions of the desirable outcomes of a game for the players) and provided tools to quantify the complexity of strategies optimal for these objectives in a set of antagonistic scenarios (the *arenas*). We assumed that strategies are implemented by finite-state machines (automaton structures with outputs). Our model was presented in Chapter 2.

In the first part, *Characterizing finite memory requirements*, we characterized several situations in which *finite memory* is sufficient to implement optimal strategies, which can be used to obtain upper bounds on the structures needed to implement them. In Chapter 4, we generalized a work about memoryless optimal strategies in finite arenas [GZ05] to a kind of finite-memory strategies. We focused on *arena-independent finite memory*, which offers a good trade-off between applicability (it encompasses all ω-regular objectives and more) and technical convenience (it allows investigating memory structures with only the given of an objective, without instantiating arenas). We recover a characterization through language-theoretic properties and a *one-to-two-player lift*, respectively discussed more extensively in Subsections 9.1.1 and 9.1.2. In Chapter 5, we provided a link between the representation of an ω-regular language as a deterministic parity automaton and a memory structure sufficient to implement optimal

[GZ05]: Gimbert et al. (2005), *Games Where You Can Play Optimally Without Any Memory*

strategies. In doing so, we showed that for a strong kind of finite-memory determinacy (*chromatic finite-memory determinacy over infinite arenas*), the ω-regular objectives are the only finite-memory-determined objectives. This result builds on previous work about memoryless strategies and prefix-independent objectives [CN06] and provides a converse to the established finite-memory determinacy of ω-regular objectives [GH82].

[CN06]: Colcombet et al. (2006), *On the positional determinacy of edge-labeled games*

[GH82]: Gurevich et al. (1982), *Trees, Automata, and Games*

In the second part, *Obtaining precise memory requirements*, we focused on various classes of ω-regular objectives (also venturing into simple topological classes). We gave precise *chromatic* memory requirements for each player, as well as ways to compute minimal memory structures. In Chapter 7, we studied objectives definable by classical deterministic finite automata and characterized the memory structures sufficient to play optimally through two decidable language-theoretic properties. We proved the NP-completeness of minimizing these memory structures. In Chapter 8, we studied objectives definable by deterministic Büchi automata and characterized the ones that admit memoryless optimal strategies, once again through decidable properties. This led to a polynomial-time algorithm for deciding the half-positionality of such objectives.

9.1.1 Links between properties of objectives

We have defined a certain number of properties of objectives throughout this thesis. In this section, our goal is to give a general overview of the links between these notions, a brief reminder of what they entail about strategy complexity, a summary of the (meta-)properties that these properties satisfy, and references to proofs and examples of these claims. We include the following properties in our comparison:

- ▶ $\mathcal{M}$-monotony (Definition 4.3.1, generalized from *monotony* [GZ05]);
- ▶ $\mathcal{M}$-selectivity (Definition 4.3.4, generalized from *selectivity* [GZ05]);
- ▶ $\mathcal{M}$-prefix-independence (Definition 5.3.1, generalized from the common *prefix-independence* notion);
- ▶ $\mathcal{M}$-cycle-consistency (Definition 5.3.3, generalized from an unnamed property in [CN06]);
- ▶ $\mathcal{M}$-strong-monotony (Definition 4.3.7, generalized from the *strong monotony* notion [BFMM11] — see also the natural reformulation in Lemma 4.3.8; it has been specialized for $\mathcal{M} = \mathcal{M}_{\mathsf{triv}}$ in Condition 1);
- ▶ $\mathcal{M}$-progress-consistency (Definition 7.4.2, which is novel as far as we know; it has been specialized for $\mathcal{M} = \mathcal{M}_{\mathsf{triv}}$ in Condition 2);
- ▶ recognizability by a DBA built on top of the prefix classifier (Condition 3, already studied for other purposes in [AFS20; AF21; BL21]).

[GZ05]: Gimbert et al. (2005), *Games Where You Can Play Optimally Without Any Memory*

[BFMM11]: Bianco et al. (2011), *Exploring the boundary of half-positionality*

[AFS20]: Angluin et al. (2020), *Polynomial Identification of ω-Automata*

[AF21]: Angluin et al. (2021), *Regular ω-languages with an informative right congruence*

[BL21]: Bohn et al. (2021), *Constructing Deterministic ω-Automata from Examples by an Extension of the RPNI Algorithm*

Stability by product. All properties that rely on a memory structure conform to the idea that adding more information is not detrimental: if an objective W is "$\mathcal{M}$-something", then for all memory structures $\mathcal{M}'$, it is also "$(\mathcal{M} \otimes \mathcal{M}')$-something". This idea was first mentioned

in Lemma 2.5.5. Here are references to proofs of these facts for each property: Lemma 4.3.6 for $\mathcal{M}$-monotony and $\mathcal{M}$-selectivity, Lemma 5.3.6 for $\mathcal{M}$-prefix-independence and $\mathcal{M}$-cycle-consistency, and Lemma 7.4.3 for $\mathcal{M}$-progress-consistency. We did not provide an explicit proof for $\mathcal{M}$-strong-monotony, but the proof would be identical to the one for $\mathcal{M}$-monotony (Lemma 4.3.6).

> **Remark 9.1.1** A slightly more general (and perhaps more elegant) argument for the observation "stability by product" would have been to consider *morphisms of memory structures*. A morphism from $\mathcal{M} = (M, m_{\mathsf{init}}, \alpha_{\mathsf{upd}})$ to $\mathcal{M}' = (M', m'_{\mathsf{init}}, \alpha'_{\mathsf{upd}})$ is a function $\phi \colon M \to M'$ such that $\phi(m_{\mathsf{init}}) = m'_{\mathsf{init}}$ and for all $m \in M$, $c \in C$, $\alpha'_{\mathsf{upd}}(\phi(m), c) = \phi(\alpha_{\mathsf{upd}}(m, c))$. The existence of a morphism from $\mathcal{M}$ to $\mathcal{M}'$ matches the intuition that $\mathcal{M}$ contains more information than $\mathcal{M}'$ and, for the above properties, we can prove that if an objective is "$\mathcal{M}'$-something", then it is also "$\mathcal{M}$-something". In particular, there is always a morphism from $\mathcal{M} \otimes \mathcal{M}'$ to $\mathcal{M}$.

Symmetry. The following properties are *symmetric* in the sense that they hold for an objective W if and only if they hold for its complement $\overline{W}$: $\mathcal{M}$-prefix-independence and $\mathcal{M}$-cycle-consistency (Lemma 5.3.6), and $\mathcal{M}$-strong-monotony (Lemma 4.3.9). All the other properties do not satisfy this, which was illustrated through examples: Remark 4.3.3 for $\mathcal{M}$-monotony, Example 4.3.5 for $\mathcal{M}$-selectivity, Example 7.4.1 and Remark 7.4.6 for $\mathcal{M}$-progress-consistency, Example 8.3.6 for DBA-recognizability by the prefix classifier. Note that $\mathcal{M}$-monotony is symmetric for ω-regular objectives (Lemma 4.3.11).

Links. We illustrate in Figure 9.1 all the links between the various properties defined in this thesis and what they were shown to imply about strategy complexity. We encompass all chapters containing original contributions. We aim for the most essential high-level results; for some of the implications, slightly stronger results were proved (e.g., (1) applies to the more general framework of *preference relations*; for (5), considering countable arenas suffices). For readability, some implications that follow by symmetry of $\mathcal{P}_1$ and $\mathcal{P}_2$ are not recalled. Implications without references follow directly from the definitions. To ease the reading, we give here the full references to proofs of the non-trivial implications from Figure 9.1:

(1) Theorem 4.4.4;
(2) Theorem 4.4.3;
(3) Theorem 5.4.4 (Conjecture 5.4.5 if we keep the same $\mathcal{M}$);
(4) Proposition 5.5.3;
(5) Proposition 5.5.5;
(6) Theorem 5.4.1 (bottom-to-top implication) and Lemma 5.3.7 (top-to-bottom implication);
(7) Folklore result explained in the proof of Theorem 2.7.11;
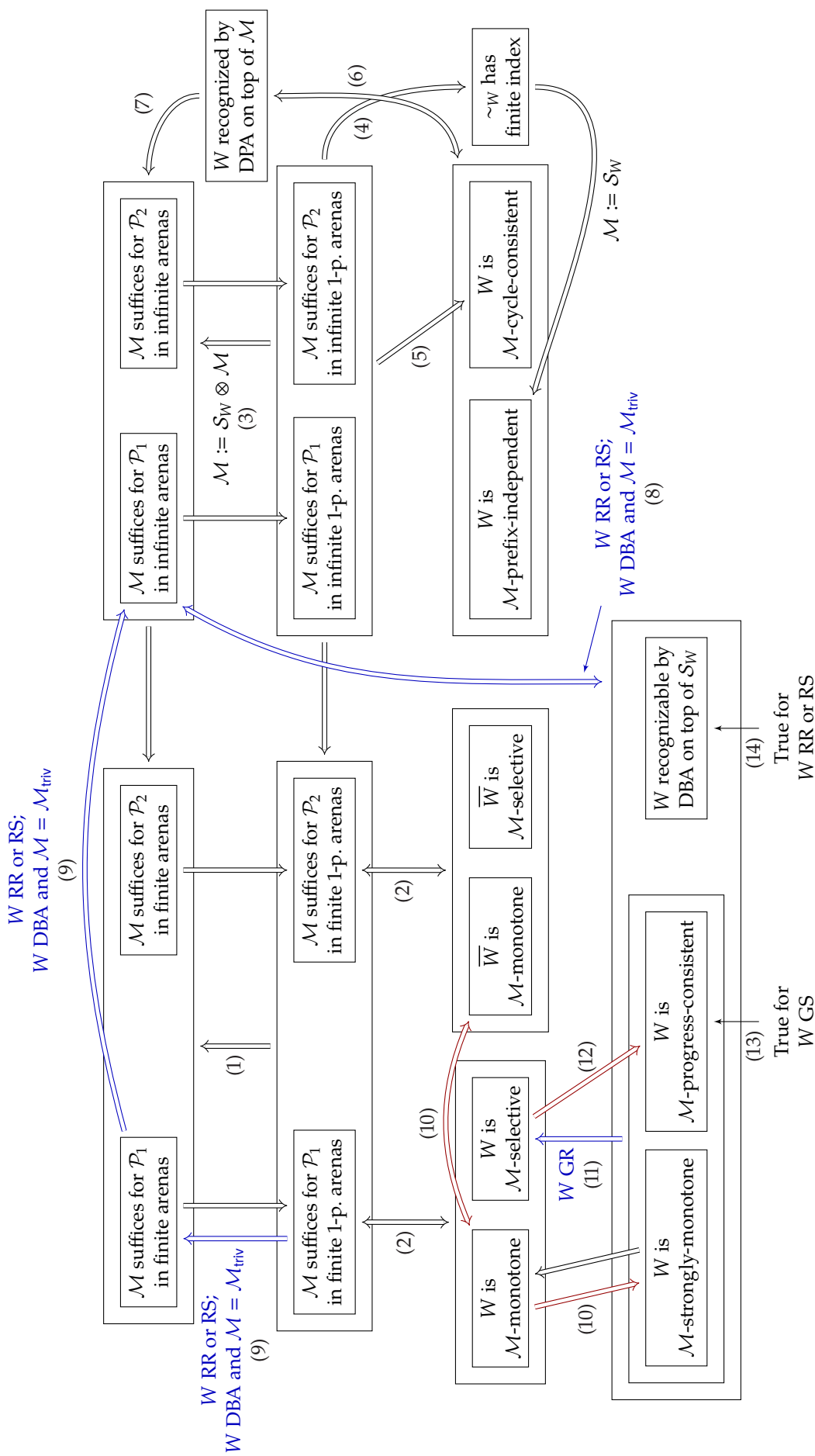
Figure 9.1: Overview of the links between many properties defined in this thesis. Let $W \subseteq C^\omega$ be an objective and $\mathcal{M}$ be a memory structure. Red implications only hold for ω-regular objectives. Blue implications hold in specific cases, which are then specified next to the implication. Notation 1-p. (resp. GR, RR, GS, RS, DBA) is used as an abbreviation for *one-player* (resp. *general reachability, regular reachability, general safety, regular safety, DBA-recognizable*).

(8) Theorem 7.4.8 (see also the close Theorem 7.4.9), Theorem 7.3.3, and Theorem 8.3.7;

(9) Theorem 7.4.14;

(10) Lemma 4.3.11 (false when W is not ω-regular in Lemma 4.3.10);

(11) Lemma 7.4.11;

(12) Lemma 7.4.12;

(13) Remark 7.4.6;

(14) Myhill-Nerode theorem [Ner58].

[Ner58]: Nerode (1958), *Linear Automaton Transformations*

Some observations were left out from the figure due to its already large size: $\mathcal{M}$-prefix-independence always implies $\mathcal{M}$-strong-monotony by definition (Remark 5.3.2), and $\mathcal{M}$-selectivity is incomparable to $\mathcal{M}$-cycle-consistency (Remark 5.3.5).

9.1.2 One-to-two-player lifts

One of the first results that we presented in this thesis was the *one-to-two-player lift for memoryless strategies in finite arenas* (Theorem 3.1.1) by Gimbert and Zielonka [GZ05]: to show that memoryless strategies suffice to play optimally in finite two-player arenas, it suffices to show it in the finite one-player arenas of both players. Generalizing and extending this result was a recurring theme throughout this thesis — we give an overview of our results through the prism of this *lift* property.

[GZ05]: Gimbert et al. (2005), *Games Where You Can Play Optimally Without Any Memory*

This result can be extended in (at least) two ways. First, by showing that the general scheme holds in other (orthogonal or more general) contexts, depending for instance on a class of arenas or a class of strategies. The meta-statement is: *if strategies X suffice for both players in their one-player arenas Y, then strategies X also suffice in two-player arenas Y*. This is what we did in Theorem 4.4.4 (generalization to arena-independent finite-memory determinacy over finite arenas) and Theorem 5.4.4 (orthogonal result: lift for chromatic finite-memory determinacy over infinite arenas). We sum up occurrences in which similar general results have been shown to hold in Figure 9.2; it turns out that this is quite a robust property. We discussed [GZ05] and [CN06; Kop08] in Chapter 3, the generalization to stochastic games in Subsection 4.8.1, and the generalization to *mildly growing memory* in Subsection 4.8.2.

[CN06]: Colcombet et al. (2006), *On the positional determinacy of edge-labeled games*
[Kop08]: Kopczyński (2008), *Half-positional Determinacy of Infinite Games*

Secondly, the lifts can also be strengthened in more restricted classes of objectives by showing stronger *asymmetric versions*: just studying the memory requirements of *one* player in its one-player games directly gives

Arenas \ Memory req.	Memoryless	Arena-independent FM	*Mildly growing*
Finite deterministic	[GZ05]	Theorem 4.4.4 [BLO+22]	[Koz22b]
Infinite deterministic	[CN06; Kop08] (*PI*)	Theorem 5.4.4 [BRV23]	
Finite stochastic	[GZ09]	[BORV21a]	

Figure 9.2: Occurrences of *one-to-two-player lifts*, when memory requirements in two-player arenas reduce to the easier question in one-player arenas. The class of strategies must suffice for *both* players. Acronym *PI* refers to the fact that this result was shown under prefix-independence hypothesis. In stochastic arenas, results deal with *pure* strategies (i.e., not resorting to randomization).

its memory requirements in two-player games. We obtained two such results. First, for *regular objectives* (both reachability and safety ones), if some memory structure suffices to play optimally in the one-player arenas of a player, it also suffices for this player in two-player arenas. Secondly, for *objectives recognizable by deterministic Büchi automata*, if memoryless strategies suffice for $\mathcal{P}_1$ in one-player arenas, then they also suffice in two-player arenas.

In both cases, we also get almost for free a *finite-to-infinite lift*: the memory requirements do not increase when going from finite to infinite arenas. This is intuitively natural for ω-regular objectives: thanks to Lemma 2.7.12 about the fact that ω-regular objectives are determined by their ultimately periodic words, we often reduced reasonings about *infinite* words and *infinite* arenas to reasoning about *ultimately periodic* words and *finite* arenas (Lemmas 4.3.11 and 7.4.12 and Propositions 7.3.1 and 7.4.4). However, we could not find a general proof of this fact, and the proofs for regular objectives and objectives recognized by DBAs required in each case to fully characterize the memory requirements through properties that are easier to manipulate.

These two observations (*asymmetric* and *finite-to-infinite* lifts for regular and DBA-recognizable objectives) spark the following conjecture about ω-regular objectives.

Conjecture 9.1.2 *Let $W \subseteq C^\omega$ be an ω-**regular** objective and $\mathcal{M}$ be a memory structure.*

1. *Asymmetric lift: if $\mathcal{M}$ suffices to play optimally for $\mathcal{P}_1$ in finite one-player arenas, then $\mathcal{M}$ suffices to play optimally for $\mathcal{P}_1$ in finite two-player arenas.*
2. *Finite-to-infinite lift: if $\mathcal{M}$ suffices to play optimally for $\mathcal{P}_1$ in finite arenas, then $\mathcal{M}$ suffices to play optimally for $\mathcal{P}_1$ in infinite arenas.*

Both items are false in general if W is not ω-regular.

1. For a non-ω-regular counterexample to the first item, we briefly reproduce an example from [Kop06, Proposition 2] (used as a counterexample to another conjecture). Let $C = \{0, 1\}$, and let $\mathsf{MP}^{\mathbb{Q}}$ denote the set of words $w \in C^\omega$ whose mean payoff is rational ($\mathsf{MP}(w) \in \mathbb{Q}$). In any finite one-player arena of $\mathcal{P}_1$, any memoryless strategy of $\mathcal{P}_1$

[Kop06]: Kopczyński (2006), *Half-Positional Determinacy of Infinite Games*
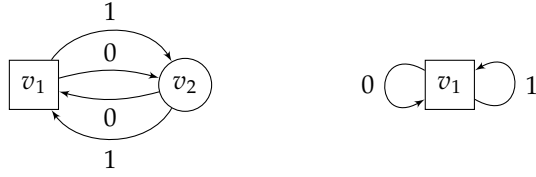
Figure 9.3: Finite two-player arena in which $\mathcal{P}_1$ needs finite memory to win for objective $\mathsf{MP}^{\mathbb{Q}}$, and finite one-player arena in which $\mathcal{P}_2$ needs infinite memory.

wins from every vertex, as it induces an ultimately periodic word, which always has a rational mean payoff. Hence, memoryless strategies suffice for $\mathcal{P}_1$ for $\mathsf{MP}^{\mathbb{Q}}$ in finite one-player arenas. Yet, $\mathcal{P}_1$ needs finite memory to win in the finite two-player arena of Figure 9.3 (left): $\mathcal{P}_1$ can guarantee a mean payoff of value $\frac{1}{2} \in \mathbb{Q}$ by replying with 1 if $\mathcal{P}_2$ played 0 and with 0 if $\mathcal{P}_2$ played 1. However, any memoryless strategy of $\mathcal{P}_1$ loses because $\mathcal{P}_2$ can induce a non-rational mean payoff without paying attention to the constant choice of $\mathcal{P}_1$. Note that objective $\mathsf{MP}^{\mathbb{Q}}$ is not finite-memory-determined: already in a finite one-player arena (e.g., the one on the right of Figure 9.3), $\mathcal{P}_2$ needs infinite memory to induce a non-rational mean payoff.

2. For a non-ω-regular counterexample to the second item, we simply take the mean-payoff objective $\mathsf{MP}^{\geq 0}$. It is memoryless-determined over finite arenas (Theorem 2.6.4, Example 3.1.2), but requires infinite memory in some infinite arena (even a countable, finitely branching, one-player arena with finitely many colors; see Example 3.2.1).

Note that even the memoryless case ($\mathcal{M} = \mathcal{M}_{\text{triv}}$) of Conjecture 9.1.2, which is conceptually simpler, is open. If both its items are taken together, this conjecture is stronger than Conjecture 5.4.5 discussed in Chapter 5.

9.2 Future prospects

9.2.1 Arena-dependent memory requirements

All the results in this thesis can be used to help determine *arena-independent* memory requirements of various classes of games. We gave multiple arguments to justify that this is a reasonable restriction: the well-known ω-regular objectives enjoy arena-independent finite-memory determinacy (Theorem 2.7.11), there is no general one-to-two-player lift in this case (Proposition 3.1.3), and over infinite arenas, we do not know of a distinction between arena-independent finite-memory determinacy and finite-memory determinacy (Remark 2.6.10).

Yet, at least when looking at finite arenas, (*arena-dependent*) finite-memory determinacy of an objective may already lead to algorithms for solving the games (e.g., in energy-parity games [CD12], multi-energy games [CRR14; JLS15], window objectives [CDRR15], average-energy games [BMR+18; BHM+17]). It would therefore make sense to go further in the understanding of this general finite-memory determinacy over games played on finite arenas. We discussed one general result about this topic in Subsection 4.8.2,

[CD12]: Chatterjee et al. (2012), *Energy parity games*

[CRR14]: Chatterjee et al. (2014), *Strategy synthesis for multi-dimensional quantitative objectives*

[JLS15]: Jurdziński et al. (2015), *Fixed-Dimensional Energy Games are in Pseudo-Polynomial Time*

[CDRR15]: Chatterjee et al. (2015), *Looking at mean-payoff and total-payoff through windows*

[BMR+18]: Bouyer et al. (2018), *Average-energy games*

[BHM+17]: Bouyer et al. (2017), *Bounding Average-Energy Games*

extending our one-to-two-player lift from arena-independent to *mildly growing* memory requirements [Koz22b].

Another line of work started by giving sufficient conditions for half-positionality over finite arenas in deterministic games [Kop06; BFMM11], Markov decision processes [Gim07], and stochastic games [GK14] through two conditions (which are roughly similar in all these works). The two conditions, as often, deal respectively with prefixes and with cycles: the first condition is prefix-independence (slightly weakened to $\mathcal{M}_{triv}$-strong-monotony in [BFMM11]); the second condition is called *concavity* (or *submixingness*), and states that two infinite losing words cannot be combined into a winning word. Half-positionality is inherently *arena-independent* (for one player). However, in [MSTW21] and a version of [GK14] updated in 2021, a corollary about finite-memory determinacy was stated: under prefix-independence and concavity, if additionally, $\mathcal{P}_2$ has finite-memory optimal strategies in its finite one-player arenas, then $\mathcal{P}_2$ has finite-memory optimal strategies in finite two-player arenas. We sum up the situation: under strong conditions for $\mathcal{P}_1$ (sufficient but not necessary for half-positionality in finite arenas), we have a one-to-two-player lift for (arena-dependent!) finite-memory determinacy of $\mathcal{P}_2$. This is strong enough to show finite-memory determinacy of multi-energy games (which satisfy concavity for one of the players and prefix-independence).

This suggests that there is yet another road for a reasonable one-to-two-player lift leading to finite-memory determinacy: what if we make a strong assumption on one player (e.g., memoryless optimal strategies in one-player arenas) but assume only the existence of finite-memory optimal strategies for the other player in one-player arenas? This leads to the following conjecture, stronger than [MSTW21; GK14] when considering deterministic games.

> **Conjecture 9.2.1** *Let $W \subseteq C^\omega$ be an objective. If $\mathcal{P}_1$ has memoryless optimal strategies in its finite one-player arenas and $\mathcal{P}_2$ has finite-memory optimal strategies in its finite one-player arenas, then W is finite-memory-determined over finite (two-player) arenas.*

This question is one of the first ones we looked at while preparing this thesis, but to no avail.

9.2.2 Chaotic memory

In Subsection 2.6.2, we defined *chaotic memory structures* (sometimes called *general* in the literature) as memory structures that can observe precise *edges* taken while playing games, and not solely their color. For this reason, chaotic memory is intrinsically *arena-dependent*, which relates it to the previous section.

[Koz22b]: Kozachinskiy (2022), *One-To-Two-Player Lifting for Mildly Growing Memory*

[Kop06]: Kopczyński (2006), *Half-Positional Determinacy of Infinite Games*

[BFMM11]: Bianco et al. (2011), *Exploring the boundary of half-positionality*

[Gim07]: Gimbert (2007), *Pure Stationary Optimal Strategies in Markov Decision Processes*

[GK14]: Gimbert et al. (2014), *Submixing and Shift-Invariant Stochastic Games*

[MSTW21]: Mayr et al. (2021), *Simple Stochastic Games with Almost-Sure Energy-Parity Objectives are in NP and coNP*

Yet, studying chaotic memory may be of interest even for objectives that are known to be arena-independent-finite-memory-determined, such as ω-regular objectives. A recent breakthrough by Casares, Colcombet, and Lehtinen [Cas22; CCL22] shows that for the class of Muller conditions, chaotic memory structures actually correspond to *good-for-games Rabin automata* recognizing the condition. Moreover, these chaotic memory structures may be exponentially smaller than the smallest chromatic memory structure.

Good-for-games Rabin automata are a kind of non-deterministic automata. As for chromatic memory structures, they read colors, but have some well-behaved non-determinism: the idea is that while playing with such a non-deterministic memory structure, the player should be able to resolve the non-determinism on the fly to its advantage. This provides an arena-independent version of a possibly more succinct memory model: what if instead of finding the smallest possible chaotic memory structure (which depends on an arena), we had to find the smallest *non-deterministic* chromatic memory structure (which depends only on the objective), with some constraint on the non-determinism? This alternative point of view is more amenable to the kind of arguments we used throughout this thesis; for example, we could define a non-deterministic chromatic memory structure $\mathcal{M}$ and look at language-theoretic properties of W and $\mathcal{M}$ (is W "$\mathcal{M}$-something"?).

Chaotic memory is therefore fully understood for Muller conditions, and also for general safety objectives [CFH14]. Given these results and the new results from this thesis, the most pressing open question is about regular and general reachability objectives (Chapter 7): how to characterize the amount of *chaotic* memory necessary and sufficient to play optimally for these objectives, how hard is it to compute, and can it be significantly smaller than the smallest (deterministic) chromatic memory structure?

9.2.3 Alternative models

We studied the classical model of two-player zero-sum turn-based games on graphs; for more fine-grained synthesis endeavors, it is worthwhile to consider other models. We mention (non-exhaustively) some alternative models that have recently been scrutinized in the literature.

Stochastic games. We discussed *stochastic games* in Subsection 4.8.1, in which probabilities are used to model some features of the interaction between the two players. An additional kind of strategy complexity to consider is *randomization*; resorting to randomness in strategies may decrease the memory requirements [CdH04; Hor09; CRR14; MPR20; MR22]. We refer to Subsection 4.8.1 for a more complete discussion.

Another frequent stochastic model is a *Markov decision process* (i.e., a one-player stochastic game). Memory requirements of finite Markov decision

These results have also been discussed with precise statements in Section 6.2.

[Cas22]: Casares (2022), *On the Minimisation of Transition-Based Rabin Automata and the Chromatic Memory Requirements of Muller Conditions*

[CCL22]: Casares et al. (2022), *On the Size of Good-For-Games Rabin Automata and Its Link with the Memory in Muller Games*

[CFH14]: Colcombet et al. (2014), *Playing Safe*

[CdH04]: Chatterjee et al. (2004), *Trading Memory for Randomness*

[Hor09]: Horn (2009), *Random Fruits on the Zielonka Tree*

[CRR14]: Chatterjee et al. (2014), *Strategy synthesis for multi-dimensional quantitative objectives*

[MPR20]: Monmege et al. (2020), *Reaching Your Goal Optimally by Playing at Random with No Memory*

[MR22]: Main et al. (2022), *Different Strokes in Randomised Strategies: Revisiting Kuhn's Theorem Under Finite-Memory Assumptions*

processes have been extensively studied (e.g., in [Gim07; DKQR20]). Recently, a new line of work looked at strategy complexity in *countable* Markov decision processes for many classical objectives (see survey [KMS⁺20] and some subsequent papers [KMST21; MM21]). It goes beyond simply distinguishing memoryless, finite-memory, and infinite-memory strategies; when infinite memory is needed, it considers the kinds of infinite memory structures that suffice to play optimally. For instance, *Markov strategies* can base their decision on the current arena vertex and on the number of steps since the start of the game. Such strategies cannot be implemented with finite memory, but they use an arguably simple and well-behaved infinite structure.

Concurrent games. We studied *turn-based games*, in which players take turns making decisions; in *concurrent games*, players make decisions simultaneously and the color seen and transition taken depend on the decisions made by both players. Concurrent games were present at the genesis of game theory, where *simultaneous games* were the model of choice for the seminal papers of von Neumann [von28] (about zero-sum games) and Nash [Nas51] (about non-zero-sum games). Concurrent games require randomization of strategies even for simple objectives [Mar98; dHK07]. There have been studies of the strategy complexity of concurrent games on graphs for classical objectives [BBL22b; BBL22a]. Our work in Chapter 4 was also extended to a well-behaved kind of concurrent games on graphs [BBL21].

Article [BLT22] establishes the existence of finite-memory optimal strategies from topological properties of objectives for concurrent games. Although general reachability and safety objectives fit into their framework, there are major differences with our work beyond concurrency: their games are not played *on graphs*. Arenas can be considered, but they have to be encoded into the objective, and strategies are therefore not assumed to be able to observe the current arena vertex (to use this information, strategies have to add it to their memory structure).

More expressive strategy models. Throughout this thesis, our finite-memory strategies are represented by deterministic finite-state machines. This classical model makes perfect sense for ω-regular objectives, thanks to their finite-memory determinacy. However, it can be seen as quite restrictive for many other behaviors; for instance, basing the decisions of a strategy on a counter taking values in the natural numbers does not fall into this scope of finite-memory strategies. Just as we gave tools to understand finite-memory determinacy using finite-state machines as strategies, it would be interesting to understand which objectives admit optimal strategies based on more expressive kinds of machines. We have already mentioned above *good-for-games* automata (which are roughly restricted *non*-deterministic finite-state machines), randomness in strategies, and some restricted kinds of infinite-memory strategies

[Gim07]: Gimbert (2007), *Pure Stationary Optimal Strategies in Markov Decision Processes*

[DKQR20]: Delgrange et al. (2020), *Simple Strategies in Multi-Objective MDPs*

[KMS⁺20]: Kiefer et al. (2020), *How to Play in Infinite MDPs (Invited Talk)*

[KMST21]: Kiefer et al. (2021), *Transience in Countable MDPs*

[MM21]: Mayr et al. (2021), *Strategy Complexity of Mean Payoff, Total Payoff and Point Payoff Objectives in Countable MDPs*

[von28]: von Neumann (1928), *Zur Theorie der Gesellschaftsspiele*

[Nas51]: Nash (1951), *Non-cooperative Games*

[Mar98]: Martin (1998), *The Determinacy of Blackwell Games*

[dHK07]: de Alfaro et al. (2007), *Concurrent reachability games*

[BBL22b]: Bordais et al. (2022), *Playing (Almost-)Optimally in Concurrent Büchi and Co-Büchi Games*

[BBL22a]: Bordais et al. (2022), *Optimal Strategies in Concurrent Reachability Games*

[BBL21]: Bordais et al. (2021), *From Local to Global Determinacy in Concurrent Graph Games*

[BLT22]: Bouyer et al. (2022), *Finite-Memory Strategies in Two-Player Infinite Games*

(e.g., *Markov strategies* for stochastic games). We mention additionally (and non-exhaustively) works on representing strategies as pushdown automata [Wal01], Turing machines [Gel14], decision trees [BCKT18], MSO-transducers [BT22], and register automata [EFLR22].

[Wal01]: Walukiewicz (2001), *Pushdown Processes: Games and Model-Checking*

[Gel14]: Gelderie (2014), *Strategy machines: representation and complexity of strategies in infinite games*

[BCKT18]: Brázdil et al. (2018), *Strategy Representation by Decision Trees in Reactive Synthesis*

[BT22]: Brütsch et al. (2022), *Solving Infinite Games in the Baire Space*

[EFLR22]: Exibard et al. (2022), *Computability of Data-Word Transductions over Different Data Domains*

Bibliography

[AF21]   Dana Angluin and Dana Fisman. 'Regular ω-languages with an informative right congruence'. In: *Information and Computation* 278 (2021), p. 104598. DOI: `10.1016/j.ic.2020.104598` (cited on pages 13, 101, 111, 196, 232).

[AFS20]   Dana Angluin, Dana Fisman, and Yaara Shoval. 'Polynomial Identification of ω-Automata'. In: *Proceedings (Part II) of the 26th International Conference on Tools and Algorithms for the Construction and Analysis of Systems, TACAS 2020, Held as Part of ETAPS 2020, Dublin, Ireland, April 25–30, 2020.* Ed. by Armin Biere and David Parker. Vol. 12079. Lecture Notes in Computer Science. Springer, 2020, pp. 325–343. DOI: `10.1007/978-3-030-45237-7_20` (cited on pages 13, 101, 111, 196, 232).

[AK19]   Bader Abu Radi and Orna Kupferman. 'Minimizing GFG Transition-Based Automata'. In: *Proceedings of the 46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, Patras, Greece, July 9–12, 2019.* Ed. by Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi. Vol. 132. LIPIcs. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019, 100:1–100:16. DOI: `10.4230/LIPIcs.ICALP.2019.100` (cited on pages 196, 198).

[AK20]   Bader Abu Radi and Orna Kupferman. 'Canonicity in GFG and Transition-Based Automata'. In: *Proceedings of the 11th International Symposium on Games, Automata, Logics, and Formal Verification, GandALF 2020, Brussels, Belgium, September 21–22, 2020.* Ed. by Jean-François Raskin and Davide Bresolin. Vol. 326. EPTCS. 2020, pp. 199–215. DOI: `10.4204/EPTCS.326.13` (cited on pages 196, 198).

[AK22]   Bader Abu Radi and Orna Kupferman. 'Minimization and Canonization of GFG Transition-Based Automata'. In: *Logical Methods in Computer Science* 18.3 (2022). DOI: `10.46298/lmcs-18(3:16)2022` (cited on pages 37, 196).

[AR17]   Benjamin Aminof and Sasha Rubin. 'First-cycle games'. In: *Information and Computation* 254 (2017), pp. 195–216. DOI: `10.1016/j.ic.2016.10.008` (cited on page 8).

[BBE10]   Tomáš Brázdil, Václav Brožek, and Kousha Etessami. 'One-Counter Stochastic Games'. In: *Proceedings of the 30th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2010, Chennai, India, December 15–18, 2010.* Ed. by Kamal Lodaya and Meena Mahajan. Vol. 8. LIPIcs. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2010, pp. 108–119. DOI: `10.4230/LIPIcs.FSTTCS.2010.108` (cited on page 112).

[BBL21]   Benjamin Bordais, Patricia Bouyer, and Stéphane Le Roux. 'From Local to Global Determinacy in Concurrent Graph Games'. In: *Proceedings of the 41st IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2021, Virtual Conference, December 15–17, 2021.* Ed. by Mikołaj Bojańczyk and Chandra Chekuri. Vol. 213. LIPIcs. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021, 41:1–41:14. DOI: `10.4230/LIPIcs.FSTTCS.2021.41` (cited on page 240).

[BBL22a]   Benjamin Bordais, Patricia Bouyer, and Stéphane Le Roux. 'Optimal Strategies in Concurrent Reachability Games'. In: *Proceedings of the 30th EACSL Annual Conference on Computer Science Logic, CSL 2022, Göttingen, Germany, February 14–19, 2022.* Ed. by

Florin Manea and Alex Simpson. Vol. 216. LIPIcs. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022, 7:1–7:17. DOI: `10.4230/LIPIcs.CSL.2022.7` (cited on page 240).

[BBL22b]   Benjamin Bordais, Patricia Bouyer, and Stéphane Le Roux. 'Playing (Almost-)Optimally in Concurrent Büchi and Co-Büchi Games'. In: *Proceedings of the 42nd IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2022, IIT Madras, Chennai, India, December 18–20, 2022.* Ed. by Anuj Dawar and Venkatesan Guruswami. Vol. 250. LIPIcs. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022, 33:1–33:18. DOI: `10.4230/LIPIcs.FSTTCS.2022.33` (cited on page 240).

[BCJ18]   Roderick Bloem, Krishnendu Chatterjee, and Barbara Jobstmann. 'Graph Games and Reactive Synthesis'. In: *Handbook of Model Checking.* Ed. by Edmund M. Clarke, Thomas A. Henzinger, Helmut Veith, and Roderick Bloem. Springer, 2018, pp. 921–962. DOI: `10.1007/978-3-319-10575-8_27` (cited on pages 5, 7, 29, 152, 195).

[BCKT18]   Tomáš Brázdil, Krishnendu Chatterjee, Jan Křetínský, and Viktor Toman. 'Strategy Representation by Decision Trees in Reactive Synthesis'. In: *Proceedings (Part I) of the 24th International Conference on Tools and Algorithms for the Construction and Analysis of Systems, TACAS 2018, Held as Part of ETAPS 2018, Thessaloniki, Greece, April 14–20, 2018.* Ed. by Dirk Beyer and Marieke Huisman. Vol. 10805. Lecture Notes in Computer Science. Springer, 2018, pp. 385–407. DOI: `10.1007/978-3-319-89960-2_21` (cited on page 241).

[BCRV22]   Patricia Bouyer, Antonio Casares, Mickael Randour, and Pierre Vandenhove. 'Half-Positional Objectives Recognized by Deterministic Büchi Automata'. In: *Proceedings of the 33rd International Conference on Concurrency Theory, CONCUR 2022, Warsaw, Poland, September 12–16, 2022.* Ed. by Bartek Klin, Sławomir Lasota, and Anca Muscholl. Vol. 243. LIPIcs. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022, 20:1–20:18. DOI: `10.4230/LIPIcs.CONCUR.2022.20` (cited on pages 15, 193).

[BCV18]   Suguman Bansal, Swarat Chaudhuri, and Moshe Y. Vardi. 'Comparator automata in quantitative verification'. In: *CoRR* abs/1812.06569 (2018). DOI: `10.48550/arXiv.1812.06569` (cited on page 138).

[BD14]   Souheib Baarir and Alexandre Duret-Lutz. 'Mechanizing the Minimization of Deterministic Generalized Büchi Automata'. In: *Proceedings of the 34th IFIP WG 6.1 International Conference on Formal Techniques for Distributed Objects, Components, and Systems, FORTE 2014, Held as Part of DisCoTec 2014, Berlin, Germany, June 3–5, 2014.* Ed. by Erika Ábrahám and Catuscia Palamidessi. Vol. 8461. Lecture Notes in Computer Science. Springer, 2014, pp. 266–283. DOI: `10.1007/978-3-662-43613-4_17` (cited on page 191).

[BFL⁺08]   Patricia Bouyer, Ulrich Fahrenberg, Kim G. Larsen, Nicolas Markey, and Jirí Srba. 'Infinite Runs in Weighted Timed Automata with Energy Constraints'. In: *Proceedings of the 6th International Conference on Formal Modeling and Analysis of Timed Systems, FORMATS 2008, Saint Malo, France, September 15–17, 2008.* Ed. by Franck Cassez and Claude Jard. Vol. 5215. Lecture Notes in Computer Science. Springer, 2008, pp. 33–47. DOI: `10.1007/978-3-540-85778-5_4` (cited on page 57).

[BFMM11] Alessandro Bianco, Marco Faella, Fabio Mogavero, and Aniello Murano. 'Exploring the boundary of half-positionality'. In: *Annals of Mathematics and Artificial Intelligence* 62.1-2 (2011), pp. 55–77. DOI: 10.1007/s10472-011-9250-1 (cited on pages 8, 13, 71, 94, 108, 194, 195, 202, 232, 238).

[BFRV22] Patricia Bouyer, Nathanaël Fijalkow, Mickael Randour, and Pierre Vandenhove. 'How to Play Optimally for Regular Objectives?' In: *CoRR* abs/2210.09703 (2022). DOI: 10.48550/arXiv.2210.09703 (cited on pages 15, 155).

[BHM⁺17] Patricia Bouyer, Piotr Hofman, Nicolas Markey, Mickael Randour, and Martin Zimmermann. 'Bounding Average-Energy Games'. In: *Proceedings of the 20th International Conference on Foundations of Software Science and Computation Structures, FoSSaCS 2017, Held as Part of ETAPS 2017, Uppsala, Sweden, April 22–29, 2017.* Ed. by Javier Esparza and Andrzej S. Murawski. Vol. 10203. Lecture Notes in Computer Science. 2017, pp. 179–195. DOI: 10.1007/978-3-662-54458-7_11 (cited on pages 8, 49, 56, 57, 237).

[BHO15] Udi Boker, Thomas A. Henzinger, and Jan Otop. 'The Target Discounted-Sum Problem'. In: *Proceedings of the 30th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2015, Kyoto, Japan, July 6–10, 2015.* IEEE Computer Society, 2015, pp. 750–761. DOI: 10.1109/LICS.2015.74 (cited on page 139).

[BHR16] Véronique Bruyère, Quentin Hautem, and Mickael Randour. 'Window parity games: an alternative approach toward parity games with time bounds'. In: *Proceedings of the 7th International Symposium on Games, Automata, Logics, and Formal Verification, GandALF 2016, Catania, Italy, September 14–16, 2016.* Ed. by Domenico Cantone and Giorgio Delzanno. Vol. 226. EPTCS. 2016, pp. 135–148. DOI: 10.4204/EPTCS.226.10 (cited on page 194).

[BHRR19] Véronique Bruyère, Quentin Hautem, Mickael Randour, and Jean-François Raskin. 'Energy Mean-Payoff Games'. In: *Proceedings of the 30th International Conference on Concurrency Theory, CONCUR 2019, Amsterdam, the Netherlands, August 27–30, 2019.* Ed. by Wan J. Fokkink and Rob van Glabbeek. Vol. 140. LIPIcs. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019, 21:1–21:17. DOI: 10.4230/LIPIcs.CONCUR.2019.21 (cited on pages 49, 56, 194).

[BJP⁺12] Roderick Bloem, Barbara Jobstmann, Nir Piterman, Amir Pnueli, and Yaniv Sa'ar. 'Synthesis of Reactive(1) designs'. In: *Journal of Computer and System Sciences* 78.3 (2012), pp. 911–938. DOI: 10.1016/j.jcss.2011.08.007 (cited on page 5).

[BK08] Christel Baier and Joost-Pieter Katoen. *Principles of model checking*. MIT Press, 2008 (cited on pages 1, 5, 39, 203).

[BL21] León Bohn and Christof Löding. 'Constructing Deterministic ω-Automata from Examples by an Extension of the RPNI Algorithm'. In: *Proceedings of the 46th International Symposium on Mathematical Foundations of Computer Science, MFCS 2021, Tallinn, Estonia, August 23–27, 2021.* Ed. by Filippo Bonchi and Simon J. Puglisi. Vol. 202. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021, 20:1–20:18. DOI: 10.4230/LIPIcs.MFCS.2021.20 (cited on pages 101, 111, 196, 232).

[BL69] J. Richard Büchi and Lawrence H. Landweber. 'Definability in the Monadic Second-Order Theory of Successor'. In: *Journal of Symbolic Logic* 34.2 (1969), pp. 166–170. DOI: 10.2307/2271090 (cited on pages 5, 11, 39, 100).

[BLO+20]   Patricia Bouyer, Stéphane Le Roux, Youssouf Oualhadj, Mickael Randour, and Pierre Vandenhove. 'Games Where You Can Play Optimally with Arena-Independent Finite Memory'. In: *Proceedings of the 31st International Conference on Concurrency Theory, CONCUR 2020, Vienna, Austria, September 1–4, 2020*. Ed. by Igor Konnov and Laura Kovács. Vol. 171. LIPIcs. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020, 24:1–24:22. DOI: `10.4230/LIPIcs.CONCUR.2020.24` (cited on pages 14, 15, 55).

[BLO+22]   Patricia Bouyer, Stéphane Le Roux, Youssouf Oualhadj, Mickael Randour, and Pierre Vandenhove. 'Games Where You Can Play Optimally with Arena-Independent Finite Memory'. In: *Logical Methods in Computer Science* 18.1 (2022). DOI: `10.46298/lmcs-18(1:11)2022` (cited on pages 14, 15, 55, 236).

[BLT22]   Patricia Bouyer, Stéphane Le Roux, and Nathan Thomasset. 'Finite-Memory Strategies in Two-Player Infinite Games'. In: *Proceedings of the 30th EACSL Annual Conference on Computer Science Logic, CSL 2022, Göttingen, Germany, February 14–19, 2022*. Ed. by Florin Manea and Alex Simpson. Vol. 216. LIPIcs. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022, 8:1–8:16. DOI: `10.4230/LIPIcs.CSL.2022.8` (cited on pages 57, 240).

[BMR+18]   Patricia Bouyer, Nicolas Markey, Mickael Randour, Kim G. Larsen, and Simon Laursen. 'Average-energy games'. In: *Acta Informatica* 55.2 (2018), pp. 91–127. DOI: `10.1007/s00236-016-0274-1` (cited on pages 8, 49, 56, 57, 237).

[Bok18]   Udi Boker. 'Why These Automata Types?' In: *Proceedings of the 22nd International Conference on Logic for Programming, Artificial Intelligence and Reasoning, LPAR 2022, Awassa, Ethiopia, November 16–21, 2018*. Ed. by Gilles Barthe, Geoff Sutcliffe, and Margus Veanes. Vol. 57. EPiC Series in Computing. EasyChair, 2018, pp. 143–163. DOI: `10.29007/c3bj` (cited on page 39).

[Boo78]   Kellogg S. Booth. 'Isomorphism Testing for Graphs, Semigroups, and Finite Automata Are Polynomially Equivalent Problems'. In: *SIAM Journal on Computing* 7.3 (1978), pp. 273–279. DOI: `10.1137/0207023` (cited on page 185).

[BORV21a]   Patricia Bouyer, Youssouf Oualhadj, Mickael Randour, and Pierre Vandenhove. 'Arena-Independent Finite-Memory Determinacy in Stochastic Games'. In: *Proceedings of the 32nd International Conference on Concurrency Theory, CONCUR 2021, Virtual Conference, August 24–27, 2021*. Ed. by Serge Haddad and Daniele Varacca. Vol. 203. LIPIcs. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021, 26:1–26:18. DOI: `10.4230/LIPIcs.CONCUR.2021.26` (cited on pages 15, 58, 96–98, 236).

[BORV21b]   Patricia Bouyer, Youssouf Oualhadj, Mickael Randour, and Pierre Vandenhove. 'Arena-Independent Finite-Memory Determinacy in Stochastic Games'. In: *CoRR* abs/2102.10104 (2021). DOI: `10.48550/arXiv.2102.10104` (cited on page 97).

[BRV22a]   Patricia Bouyer, Mickael Randour, and Pierre Vandenhove. 'Characterizing Omega-Regularity Through Finite-Memory Determinacy of Games on Infinite Graphs'. In: *Proceedings of the 39th International Symposium on Theoretical Aspects of Computer Science, STACS 2022, Marseille, France, March 15–18, 2022*. Ed. by Petra Berenbrink and Benjamin Monmege. Vol. 219. LIPIcs. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022, 16:1–16:16. DOI: `10.4230/LIPIcs.STACS.2022.16` (cited on pages 15, 100).

[BRV22b]   Patricia Bouyer, Mickael Randour, and Pierre Vandenhove. 'The True Colors of Memory: A Tour of Chromatic-Memory Strategies in Zero-Sum Games on Graphs (Invited Talk)'. In: *Proceedings of the 42nd IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2022, IIT Madras, Chennai, India, December 18–20, 2022.* Ed. by Anuj Dawar and Venkatesan Guruswami. Vol. 250. LIPIcs. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022, 3:1–3:18. DOI: `10.4230/LIPIcs.FSTTCS.2022.3` (cited on page 15).

[BRV23]   Patricia Bouyer, Mickael Randour, and Pierre Vandenhove. 'Characterizing Omega-Regularity Through Finite-Memory Determinacy of Games on Infinite Graphs'. In: *TheoretiCS* 2 (2023), pp. 1–48. DOI: `10.46298/theoretics.23.1` (cited on pages 15, 100, 236).

[BT22]   Benedikt Brütsch and Wolfgang Thomas. 'Solving Infinite Games in the Baire Space'. In: *Fundamenta Informaticae* 186.1-4 (2022), pp. 63–88. DOI: `10.3233/FI-222119` (cited on page 241).

[Büc60]   J. Richard Büchi. 'Weak Second-Order Arithmetic and Finite Automata'. In: *Mathematical Logic Quarterly* 6.1–6 (1960), pp. 66–92. DOI: `10.1002/malq.19600060105` (cited on page 5).

[Büc62]   J. Richard Büchi. 'On a Decision Method in Restricted Second Order Arithmetic'. In: *Proceedings of the International Congress on Logic, Methodology and Philosophy of Science* (1962), pp. 1–11. DOI: `10.1007/978-1-4613-8928-6_23` (cited on page 5).

[Cas22]   Antonio Casares. 'On the Minimisation of Transition-Based Rabin Automata and the Chromatic Memory Requirements of Muller Conditions'. In: *Proceedings of the 30th EACSL Annual Conference on Computer Science Logic, CSL 2022, Göttingen, Germany, February 14–19, 2022.* Ed. by Florin Manea and Alex Simpson. Vol. 216. LIPIcs. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022, 12:1–12:17. DOI: `10.4230/LIPIcs.CSL.2022.12` (cited on pages 12, 29, 30, 37, 57, 147, 151, 152, 162, 221, 239).

[CCF21]   Antonio Casares, Thomas Colcombet, and Nathanaël Fijalkow. 'Optimal Transformations of Games and Automata Using Muller Conditions'. In: *Proceedings of the 48th International Colloquium on Automata, Languages, and Programming, ICALP 2021, Glasgow, Scotland, July 12–16, 2021.* Ed. by Nikhil Bansal, Emanuela Merelli, and James Worrell. Vol. 198. LIPIcs. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021, 123:1–123:14. DOI: `10.4230/LIPIcs.ICALP.2021.123` (cited on pages 118, 152).

[CCL22]   Antonio Casares, Thomas Colcombet, and Karoliina Lehtinen. 'On the Size of Good-For-Games Rabin Automata and Its Link with the Memory in Muller Games'. In: *Proceedings of the 49th International Colloquium on Automata, Languages, and Programming, ICALP 2022, Paris, France, July 4–8, 2022.* Ed. by Mikołaj Bojańczyk, Emanuela Merelli, and David P. Woodruff. Vol. 229. LIPIcs. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022, 117:1–117:20. DOI: `10.4230/LIPIcs.ICALP.2022.117` (cited on pages 12, 30, 152, 239).

[CD12]   Krishnendu Chatterjee and Laurent Doyen. 'Energy parity games'. In: *Theoretical Computer Science* 458 (2012), pp. 49–60. DOI: `10.1016/j.tcs.2012.07.038` (cited on pages 8, 194, 237).

[CD16]   Krishnendu Chatterjee and Laurent Doyen. 'Perfect-Information Stochastic Games with Generalized Mean-Payoff Objectives'. In: *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2016, New York, NY, USA, July 5–8, 2016*. Ed. by Martin Grohe, Eric Koskinen, and Natarajan Shankar. ACM, 2016, pp. 247–256. DOI: `10.1145/2933575.2934513` (cited on page 102).

[CDGH10]   Krishnendu Chatterjee, Laurent Doyen, Hugo Gimbert, and Thomas A. Henzinger. 'Randomness for Free'. In: *Proceedings of the 35th International Symposium on Mathematical Foundations of Computer Science, MFCS 2010, Brno, Czech Republic, August 23–27, 2010*. Ed. by Petr Hlinený and Antonín Kucera. Vol. 6281. Lecture Notes in Computer Science. Springer, 2010, pp. 246–257. DOI: `10.1007/978-3-642-15155-2_23` (cited on page 98).

[CdH04]   Krishnendu Chatterjee, Luca de Alfaro, and Thomas A. Henzinger. 'Trading Memory for Randomness'. In: *Proceedings of the 1st International Conference on Quantitative Evaluation of Systems, QEST 2004, Enschede, The Netherlands, 27–30 September, 2004*. IEEE Computer Society, 2004, pp. 206–217. DOI: `10.1109/QEST.2004.1348035` (cited on pages 98, 239).

[CDH09]   Krishnendu Chatterjee, Laurent Doyen, and Thomas A. Henzinger. 'Expressiveness and Closure Properties for Quantitative Languages'. In: *Proceedings of the 24th Annual IEEE Symposium on Logic in Computer Science, LICS 2009, Los Angeles, CA, USA, 11–14 August 2009*. IEEE Computer Society, 2009, pp. 199–208. DOI: `10.1109/LICS.2009.16` (cited on page 138).

[CdHS03]   Arindam Chakrabarti, Luca de Alfaro, Thomas A. Henzinger, and Mariëlle Stoelinga. 'Resource Interfaces'. In: *Proceedings of the 3rd International Conference on Embedded Software, EMSOFT 2003, Philadelphia, PA, USA, October 13–15, 2003*. Ed. by Rajeev Alur and Insup Lee. Vol. 2855. Lecture Notes in Computer Science. Springer, 2003, pp. 117–133. DOI: `10.1007/978-3-540-45212-6_9` (cited on pages 8, 34).

[CDK93]   Edmund M. Clarke, I. A. Draghicescu, and Robert P. Kurshan. 'A Unified Approch for Showing Language Inclusion and Equivalence Between Various Types of omega-Automata'. In: *Information Processing Letters* 46.6 (1993), pp. 301–308. DOI: `10.1016/0020-0190(93)90069-L` (cited on page 206).

[CDRR15]   Krishnendu Chatterjee, Laurent Doyen, Mickael Randour, and Jean-François Raskin. 'Looking at mean-payoff and total-payoff through windows'. In: *Information and Computation* 242 (2015), pp. 25–52. DOI: `10.1016/j.ic.2015.03.010` (cited on pages 8, 194, 237).

[CFGO22]   Thomas Colcombet, Nathanaël Fijalkow, Pawel Gawrychowski, and Pierre Ohlmann. 'The Theory of Universal Graphs for Infinite Duration Games'. In: *Logical Methods in Computer Science* 18.3 (2022). DOI: `10.46298/lmcs-18(3:29)2022` (cited on pages 195, 204).

[CFH14]   Thomas Colcombet, Nathanaël Fijalkow, and Florian Horn. 'Playing Safe'. In: *Proceedings of the 34th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2014, New Delhi, India, December 15–17, 2014*. Ed. by Venkatesh Raman and S. P. Suresh. Vol. 29. LIPIcs. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2014, pp. 379–390. DOI: `10.4230/LIPIcs.FSTTCS.2014.379` (cited on pages 13, 112, 156, 162, 163, 175, 192, 239).

[CFH22]   Thomas Colcombet, Nathanaël Fijalkow, and Florian Horn. 'Playing Safe, Ten Years Later'. In: *CoRR* abs/2212.12024 (2022). DOI: `10.48550/arXiv.2212.12024` (cited on pages 13, 156, 192).

[CFK+12]   Taolue Chen, Vojtech Forejt, Marta Z. Kwiatkowska, Aistis Simaitis, Ashutosh Trivedi, and Michael Ummels. 'Playing Stochastic Games Precisely'. In: *Proceedings of the 23rd International Conference on Concurrency Theory, CONCUR 2012, Newcastle, UK, September 4–7, 2012*. Ed. by Maciej Koutny and Irek Ulidowski. Vol. 7454. Lecture Notes in Computer Science. Springer, 2012, pp. 348–363. DOI: `10.1007/978-3-642-32940-1_25` (cited on page 97).

[CHJ05]   Krishnendu Chatterjee, Thomas A. Henzinger, and Marcin Jurdziński. 'Mean-Payoff Parity Games'. In: *Proceedings of the 20th Annual IEEE Symposium on Logic in Computer Science, LICS 2005, Chicago, IL, USA, June 26–29, 2005*. IEEE Computer Society, 2005, pp. 178–187. DOI: `10.1109/LICS.2005.26` (cited on page 194).

[CHP07]   Krishnendu Chatterjee, Thomas A. Henzinger, and Nir Piterman. 'Generalized Parity Games'. In: *Proceedings of the 10th International Conference on Foundations of Software Science and Computational Structures, FoSSaCS 2007, Held as Part of ETAPS 2007, Braga, Portugal, March 24 – April 1, 2007*. Ed. by Helmut Seidl. Vol. 4423. Lecture Notes in Computer Science. Springer, 2007, pp. 153–167. DOI: `10.1007/978-3-540-71389-0_12` (cited on pages 8, 57).

[Chu57]   Alonzo Church. 'Application of Recursive Arithmetic to the Problem of Circuit Synthesis'. In: *Summaries of the Summer Institute of Symbolic Logic* I (1957), pp. 3–50. DOI: `10.2307/2271310` (cited on page 5).

[CJK+17]   Cristian S. Calude, Sanjay Jain, Bakhadyr Khoussainov, Wei Li, and Frank Stephan. 'Deciding parity games in quasipolynomial time'. In: *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19–23, 2017*. Ed. by Hamed Hatami, Pierre McKenzie, and Valerie King. ACM, 2017, pp. 252–263. DOI: `10.1145/3055399.3055409` (cited on page 7).

[CMJ04]   Krishnendu Chatterjee, Rupak Majumdar, and Marcin Jurdziński. 'On Nash Equilibria in Stochastic Games'. In: *Proceedings of the 18th International Workshop on Computer Science Logic, CSL 2004, 13th Annual Conference of the EACSL, Karpacz, Poland, September 20–24, 2004*. Ed. by Jerzy Marcinkowski and Andrzej Tarlecki. Vol. 3210. Lecture Notes in Computer Science. Springer, 2004, pp. 26–40. DOI: `10.1007/978-3-540-30124-0_6` (cited on page 98).

[CN06]   Thomas Colcombet and Damian Niwiński. 'On the positional determinacy of edge-labeled games'. In: *Theoretical Computer Science* 352.1-3 (2006), pp. 190–196. DOI: `10.1016/j.tcs.2005.10.046` (cited on pages 8, 10, 11, 15, 23, 49, 53, 100, 102, 107, 110, 112, 118, 119, 147, 194, 204, 232, 235, 236).

[CO22]   Antonio Casares and Pierre Ohlmann. 'Characterising memory in infinite games'. In: *CoRR* abs/2209.12044 (2022). DOI: `10.48550/arXiv.2209.12044` (cited on pages 30, 31, 116).

[CRR14]   Krishnendu Chatterjee, Mickael Randour, and Jean-François Raskin. 'Strategy synthesis for multi-dimensional quantitative objectives'. In: *Acta Informatica* 51.3-4 (2014), pp. 129–163. DOI: `10.1007/s00236-013-0182-6` (cited on pages 8, 33, 34, 57, 98, 99, 237, 239).

[dHK07]   Luca de Alfaro, Thomas A. Henzinger, and Orna Kupferman. 'Concurrent reachability games'. In: *Theoretical Computer Science* 386.3 (2007), pp. 188–217. DOI: `10.1016/j.tcs.2007.07.008` (cited on page 240).

[Dil50]   R. P. Dilworth. 'A Decomposition Theorem for Partially Ordered Sets'. In: *Annals of Mathematics* 51.1 (1950), pp. 161–166. DOI: `10.2307/1969503` (cited on page 175).

[DJW97]   Stefan Dziembowski, Marcin Jurdziński, and Igor Walukiewicz. 'How Much Memory is Needed to Win Infinite Games?' In: *Proceedings of the 12th Annual IEEE Symposium on Logic in Computer Science, LICS 1997, Warsaw, Poland, June 29 – July 2, 1997*. IEEE Computer Society, 1997, pp. 99–110. DOI: `10.1109/LICS.1997.614939` (cited on pages 7, 12, 23, 57, 147, 151–153, 203, 221).

[DKQR20]   Florent Delgrange, Joost-Pieter Katoen, Tim Quatmann, and Mickael Randour. 'Simple Strategies in Multi-Objective MDPs'. In: *Proceedings (Part I) of the 26th International Conference on Tools and Algorithms for the Construction and Analysis of Systems, TACAS 2020, Held as Part of ETAPS 2020, Dublin, Ireland, April 25–30, 2020*. Ed. by Armin Biere and David Parker. Vol. 12078. Lecture Notes in Computer Science. Springer, 2020, pp. 346–364. DOI: `10.1007/978-3-030-45190-5_19` (cited on page 240).

[DLF⁺16]   Alexandre Duret-Lutz, Alexandre Lewkowicz, Amaury Fauchille, Thibaud Michaud, Etienne Renault, and Laurent Xu. 'Spot 2.0 — A Framework for LTL and ω-Automata Manipulation'. In: *Proceedings of the 14th International Symposium on Automated Technology for Verification and Analysis, ATVA 2016, Chiba, Japan, October 17–20, 2016*. Ed. by Cyrille Artho, Axel Legay, and Doron Peled. Vol. 9938. Lecture Notes in Computer Science. 2016, pp. 122–129. DOI: `10.1007/978-3-319-46520-3_8` (cited on page 37).

[EFLR22]   Léo Exibard, Emmanuel Filiot, Nathan Lhote, and Pierre-Alain Reynier. 'Computability of Data-Word Transductions over Different Data Domains'. In: *Logical Methods in Computer Science* 18.3 (2022). DOI: `10.46298/lmcs-18(3:9)2022` (cited on page 241).

[EJ91]   E. Allen Emerson and Charanjit S. Jutla. 'Tree Automata, Mu-Calculus and Determinacy (Extended Abstract)'. In: *Proceedings of the 32nd Annual Symposium on Foundations of Computer Science, FOCS 1991, San Juan, Puerto Rico, October, 1991*. IEEE Computer Society, 1991, pp. 368–377. DOI: `10.1109/SFCS.1991.185392` (cited on pages 7, 11, 29, 109, 111).

[Elg61]   Calvin C. Elgot. 'Decision Problems of Finite Automata Design and Related Arithmetics'. In: *Transactions of the American Mathematical Society* 98.1 (1961), pp. 21–51. DOI: `10.2307/1993511` (cited on page 5).

[EM79]   Andrzej Ehrenfeucht and Jan Mycielski. 'Positional Strategies for Mean Payoff Games'. In: *International Journal of Game Theory* 8.2 (1979), pp. 109–113. DOI: `10.1007/BF01768705` (cited on pages 8, 29, 48, 112).

[FH10]   Nathanaël Fijalkow and Florian Horn. 'The surprizing complexity of reachability games'. In: *CoRR* abs/1010.2420 (2010). DOI: `10.48550/arXiv.1010.2420` (cited on pages 68, 79, 175).

[FRS03]   Stefan Felsner, Vijay Raghavan, and Jeremy P. Spinrad. 'Recognition Algorithms for Orders of Small Width and Graphs of Small Dilworth Number'. In: *Order* 20.4 (2003), pp. 351–364. DOI: `10.1023/B:ORDE.0000034609.99940.fb` (cited on page 175).

[Gel14]   Marcus Gelderie. 'Strategy machines: representation and complexity of strategies in infinite games'. PhD thesis. RWTH Aachen University, 2014 (cited on page 241).

[GH82]   Yuri Gurevich and Leo Harrington. 'Trees, Automata, and Games'. In: *Proceedings of the 14th Annual ACM Symposium on Theory of Computing, STOC 1982, San Francisco, CA, USA, May 5–7, 1982*. Ed. by Harry R. Lewis, Barbara B. Simons, Walter A. Burkhard, and Lawrence H. Landweber. ACM, 1982, pp. 60–65. DOI: `10.1145/800070.802177` (cited on pages 5, 7, 11, 39, 100, 101, 151, 232).

[Gim07]   Hugo Gimbert. 'Pure Stationary Optimal Strategies in Markov Decision Processes'. In: *Proceedings of the 24th Annual Symposium on Theoretical Aspects of Computer Science, STACS 2007, Aachen, Germany, February 22–24, 2007*. Ed. by Wolfgang Thomas and Pascal Weil. Vol. 4393. Lecture Notes in Computer Science. Springer, 2007, pp. 200–211. DOI: `10.1007/978-3-540-70918-3_18` (cited on pages 58, 238, 240).

[GK14]   Hugo Gimbert and Edon Kelmendi. 'Submixing and Shift-Invariant Stochastic Games'. In: *CoRR* abs/1401.6575 (2014). DOI: `10.48550/arXiv.1401.6575` (cited on pages 48, 58, 102, 204, 238).

[GL02]   Dimitra Giannakopoulou and Flavio Lerda. 'From States to Transitions: Improving Translation of LTL Formulae to Büchi Automata'. In: *Proceedings of the 22nd IFIP WG 6.1 International Conference on Formal Techniques for Networked and Distributed Systems, FORTE 2002, Houston, TX, USA, November 11–14, 2002*. Ed. by Doron A. Peled and Moshe Y. Vardi. Vol. 2529. Lecture Notes in Computer Science. Springer, 2002, pp. 308–326. DOI: `10.1007/3-540-36135-9_20` (cited on page 37).

[GS53]   David Gale and F. M. Stewart. 'Infinite Games with Perfect Information'. In: *Contributions to the Theory of Games (Annals of Mathematics Studies 28)*. Ed. by Harold W. Kuhn and Albert W. Tucker. Vol. II. Princeton University Press, 1953, pp. 245–266 (cited on page 23).

[GTW02]   Erich Grädel, Wolfgang Thomas, and Thomas 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. DOI: `10.1007/3-540-36387-4` (cited on pages 5, 39).

[GW06]   Erich Grädel and Igor Walukiewicz. 'Positional Determinacy of Games with Infinitely Many Priorities'. In: *Logical Methods in Computer Science* 2.4 (2006). DOI: `10.2168/LMCS-2(4:6)2006` (cited on pages 112, 116).

[GY65]   Abraham Ginzburg and Michael Yoeli. 'Products of Automata and the Problem of Covering'. In: *Transactions of the American Mathematical Society* 116 (1965), pp. 253–266 (cited on page 173).

[GZ04]   Hugo Gimbert and Wiesław Zielonka. 'When Can You Play Positionally?' In: *Proceedings of the 29th International Symposium on Mathematical Foundations of Computer Science, MFCS 2004, Prague, Czech Republic, August 22–27, 2004*. Ed. by Jiří Fiala, Václav Koubek, and Jan Kratochvíl. Vol. 3153. Lecture Notes in Computer Science. Springer, 2004, pp. 686–697. DOI: `10.1007/978-3-540-28629-5_53` (cited on pages 8, 47, 48, 112, 146).

[GZ05]   Hugo Gimbert and Wiesław Zielonka. 'Games Where You Can Play Optimally Without Any Memory'. In: *Proceedings of the 16th International Conference on Concurrency Theory, CONCUR 2005, San Francisco, CA, USA, August 23–26, 2005*. Ed. by Martín Abadi and Luca de Alfaro. Vol. 3653. Lecture Notes in Computer Science. Springer, 2005, pp. 428–442. DOI: `10.1007/11539452_33` (cited on pages 8, 10, 11, 14, 23, 47, 48, 51, 55–58, 61, 63, 65–68, 76, 77, 194, 231, 232, 235, 236).

[GZ09]   Hugo Gimbert and Wiesław Zielonka. 'Pure and Stationary Optimal Strategies in Perfect-Information Stochastic Games with Global Preferences'. Unpublished. 2009 (cited on pages 11, 58, 97, 98, 236).

[Hor09]   Florian Horn. 'Random Fruits on the Zielonka Tree'. In: *Proceedings of the 26th International Symposium on Theoretical Aspects of Computer Science, STACS 2009, Freiburg, Germany, February 26–28, 2009*. Ed. by Susanne Albers and Jean-Yves Marion. Vol. 3. LIPIcs. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Germany, 2009, pp. 541–552. DOI: `10.4230/LIPIcs.STACS.2009.1848` (cited on pages 98, 239).

[HP06]   Thomas A. Henzinger and Nir Piterman. 'Solving Games Without Determinization'. In: *Proceedings of the 20th International Workshop on Computer Science Logic, CSL 2006, 15th Annual Conference of the EACSL, Szeged, Hungary, September 25–29, 2006*. Ed. by Zoltán Ésik. Vol. 4207. Lecture Notes in Computer Science. Springer, 2006, pp. 395–410. DOI: `10.1007/11874683_26` (cited on page 152).

[HP84]   David Harel and Amir Pnueli. 'On the Development of Reactive Systems'. In: *Proceedings of the Conference on Logics and Models of Concurrent Systems, Colle-sur-Loup, France, October 8–19, 1984*. Ed. by Krzysztof R. Apt. Vol. 13. NATO ASI Series. Springer, 1984, pp. 477–498. DOI: `10.1007/978-3-642-82453-1_17` (cited on page 1).

[IMM18]   Alexey Ignatiev, António Morgado, and João Marques-Silva. 'PySAT: A Python Toolkit for Prototyping with SAT Oracles'. In: *Proceedings of the 21st International Conference on the Theory and Applications of Satisfiability Testing, SAT 2018, Held as Part of FloC 2018, Oxford, UK, July 9–12, 2018*. Ed. by Olaf Beyersdorff and Christoph M. Wintersteiger. Vol. 10929. Lecture Notes in Computer Science. Springer, 2018, pp. 428–437. DOI: `10.1007/978-3-319-94144-8_26` (cited on pages 157, 189).

[JLS15]   Marcin Jurdziński, Ranko Lazić, and Sylvain Schmitz. 'Fixed-Dimensional Energy Games are in Pseudo-Polynomial Time'. In: *Proceedings (Part II) of the 42nd International Colloquium on Automata, Languages, and Programming, ICALP 2015, Kyoto, Japan, July 6–10, 2015*. Ed. by Magnús M. Halldórsson, Kazuo Iwama, Naoki Kobayashi, and Bettina Speckmann. Vol. 9135. Lecture Notes in Computer Science. Springer, 2015, pp. 260–272. DOI: `10.1007/978-3-662-47666-6_21` (cited on pages 8, 33, 57, 99, 237).

[Kar72]   Richard M. Karp. 'Reducibility Among Combinatorial Problems'. In: *Proceedings of a symposium on the Complexity of Computer Computations, Yorktown Heights, NY, USA, March 20–22, 1972*. Ed. by Raymond E. Miller and James W. Thatcher. The IBM Research Symposia Series. Plenum Press, New York, 1972, pp. 85–103. DOI: `10.1007/978-1-4684-2001-2_9` (cited on page 185).

[Kec95]   Alexander S. Kechris. *Classical Descriptive Set Theory*. Graduate Texts in Mathematics. Springer New York, NY, 1995 (cited on page 178).

[KK20]   Igor Konnov and Laura Kovács, eds. *Proceedings of the 31st International Conference on Concurrency Theory, CONCUR 2020, Vienna, Austria, September 1–4, 2020*. Vol. 171. LIPIcs. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020.

[KK91]   Nils Klarlund and Dexter Kozen. 'Rabin Measures and Their Applications to Fairness and Automata Theory'. In: *Proceedings of the 6th Annual IEEE Symposium on Logic in Computer Science, LICS 1991, Amsterdam, The Netherlands, July 15–18, 1991*. IEEE Computer Society, 1991, pp. 256–265. DOI: `10.1109/LICS.1991.151650` (cited on pages 12, 194).

[Kla94] Nils Klarlund. 'Progress Measures, Immediate Determinacy, and a Subset Construction for Tree Automata'. In: *Annals of Pure and Applied Logic* 69.2-3 (1994), pp. 243–268. DOI: `10.1016/0168-0072(94)90086-8` (cited on pages 152, 194).

[KMS⁺20] Stefan Kiefer, Richard Mayr, Mahsa Shirmohammadi, Patrick Totzke, and Dominik Wojtczak. 'How to Play in Infinite MDPs (Invited Talk)'. In: *Proceedings of the 47th International Colloquium on Automata, Languages, and Programming, ICALP 2020, Saarbrücken, Germany, July 8–11, 2020*. Ed. by Artur Czumaj, Anuj Dawar, and Emanuela Merelli. Vol. 168. LIPIcs. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020, 3:1–3:18. DOI: `10.4230/LIPIcs.ICALP.2020.3` (cited on pages 23, 240).

[KMS18] Jan Křetínský, Tobias Meggendorfer, and Salomon Sickert. 'Owl: A Library for ω-Words, Automata, and LTL'. In: *Proceedings of the 16th International Symposium on Automated Technology for Verification and Analysis, ATVA 2018, Los Angeles, CA, USA, October 7–10, 2018*. Ed. by Shuvendu K. Lahiri and Chao Wang. Vol. 11138. Lecture Notes in Computer Science. Springer, 2018, pp. 543–550. DOI: `10.1007/978-3-030-01090-4_34` (cited on page 37).

[KMST21] Stefan Kiefer, Richard Mayr, Mahsa Shirmohammadi, and Patrick Totzke. 'Transience in Countable MDPs'. In: *Proceedings of the 32nd International Conference on Concurrency Theory, CONCUR 2021, Virtual Conference, August 24–27, 2021*. Ed. by Serge Haddad and Daniele Varacca. Vol. 203. LIPIcs. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021, 11:1–11:15. DOI: `10.4230/LIPIcs.CONCUR.2021.11` (cited on pages 118, 240).

[Kőn27] Dénes Kőnig. 'Über eine Schlussweise aus dem Endlichen ins Unendliche'. In: *Acta Scientiarum Mathematicarum* 3 (1927), pp. 121–130 (cited on page 66).

[Kop06] Eryk Kopczyński. 'Half-Positional Determinacy of Infinite Games'. In: *Proceedings (Part II) of the 33rd International Colloquium on Automata, Languages and Programming, ICALP 2006, Venice, Italy, July 10–14, 2006*. Ed. by Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener. Vol. 4052. Lecture Notes in Computer Science. Springer, 2006, pp. 336–347. DOI: `10.1007/11787006_29` (cited on pages 8, 13, 28, 48, 58, 77, 102, 194, 195, 202, 204, 236, 238).

[Kop07] Eryk Kopczyński. 'Omega-Regular Half-Positional Winning Conditions'. In: *Proceedings of the 21st International Workshop on Computer Science Logic, CSL 2007, 16th Annual Conference of the EACSL, Lausanne, Switzerland, September 11–15, 2007*. Ed. by Jacques Duparc and Thomas A. Henzinger. Vol. 4646. Lecture Notes in Computer Science. Springer, 2007, pp. 41–53. DOI: `10.1007/978-3-540-74915-8_7` (cited on page 194).

[Kop08] Eryk Kopczyński. 'Half-positional Determinacy of Infinite Games'. PhD thesis. Warsaw University, 2008 (cited on pages 10, 11, 29, 30, 53, 75, 76, 162, 194, 235, 236).

[Koz22a] Alexander Kozachinskiy. 'Energy Games over Totally Ordered Groups'. In: *CoRR* abs/2205.04508 (2022). DOI: `10.48550/arXiv.2205.04508` (cited on page 194).

[Koz22b] Alexander Kozachinskiy. 'One-To-Two-Player Lifting for Mildly Growing Memory'. In: *Proceedings of the 39th International Symposium on Theoretical Aspects of Computer Science, STACS 2022, Marseille, France, March 15–18, 2022*. Ed. by Petra Berenbrink and Benjamin Monmege. Vol. 219. LIPIcs. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022, 43:1–43:21. DOI: `10.4230/LIPIcs.STACS.2022.43` (cited on pages 11, 58, 97–99, 236, 238).

[Koz22c]   Alexander Kozachinskiy. 'State Complexity of Chromatic Memory in Infinite-Duration Games'. In: *CoRR* abs/2201.09297 (2022). DOI: `10.48550/arXiv.2201.09297` (cited on page 30).

[KS15]   Denis Kuperberg and Michal Skrzypczak. 'On Determinisation of Good-for-Games Automata'. In: *Proceedings (Part II) of the 42nd International Colloquium on Automata, Languages, and Programming, ICALP 2015, Kyoto, Japan, July 6–10, 2015*. Ed. by Magnús M. Halldórsson, Kazuo Iwama, Naoki Kobayashi, and Bettina Speckmann. Vol. 9135. Lecture Notes in Computer Science. Springer, 2015, pp. 299–310. DOI: `10.1007/978-3-662-47666-6_24` (cited on page 198).

[LeR13]   Stéphane Le Roux. 'Infinite sequential Nash equilibrium'. In: *Logical Methods in Computer Science* 9.2 (2013). DOI: `10.2168/LMCS-9(2:3)2013` (cited on page 60).

[LeR18]   Stéphane Le Roux. 'Concurrent Games and Semi-Random Determinacy'. In: *Proceedings of the 43rd International Symposium on Mathematical Foundations of Computer Science, MFCS 2018, Liverpool, UK, August 27–31, 2018*. Ed. by Igor Potapov, Paul G. Spirakis, and James Worrell. Vol. 117. LIPIcs. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018, 40:1–40:15. DOI: `10.4230/LIPIcs.MFCS.2018.40` (cited on page 57).

[LeR20]   Stéphane Le Roux. 'Time-Aware Uniformization of Winning Strategies'. In: *Proceedings of the 16th Conference on Computability in Europe – Beyond the Horizon of Computability, CiE 2020, Fisciano, Italy, June 29 – July 3, 2020*. Ed. by Marcella Anselmo, Gianluca Della Vedova, Florin Manea, and Arno Pauly. Vol. 12098. Lecture Notes in Computer Science. Springer, 2020, pp. 193–204. DOI: `10.1007/978-3-030-51466-2_17` (cited on page 30).

[LeS90]   Bertrand Le Saëc. 'Saturating right congruences'. In: *RAIRO – Theoretical Informatics and Applications* 24 (1990), pp. 545–559. DOI: `10.1051/ita/1990240605451` (cited on page 196).

[Löd01]   Christof Löding. 'Efficient minimization of deterministic weak omega-automata'. In: *Information Processing Letters* 79.3 (2001), pp. 105–109. DOI: `10.1016/S0020-0190(00)00183-6` (cited on page 229).

[LPR18]   Stéphane Le Roux, Arno Pauly, and Mickael Randour. 'Extending Finite-Memory Determinacy by Boolean Combination of Winning Conditions'. In: *Proceedings of the 38th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2018, Ahmedabad, India, December 11–13, 2018*. Ed. by Sumit Ganguly and Paritosh K. Pandya. Vol. 122. LIPIcs. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018, 38:1–38:20. DOI: `10.4230/LIPIcs.FSTTCS.2018.38` (cited on pages 57, 157).

[Mar75]   Donald A. Martin. 'Borel determinacy'. In: *Annals of Mathematics* 102 (1975), pp. 363–371. DOI: `10.2307/1971035` (cited on pages 6, 23, 63, 97).

[Mar98]   Donald A. Martin. 'The Determinacy of Blackwell Games'. In: *Journal of Symbolic Logic* 63.4 (1998), pp. 1565–1581. DOI: `10.2307/2586667` (cited on pages 97, 240).

[Mas15]   Ayala Mashiah-Yaakovi. 'Correlated Equilibria in Stochastic Games with Borel Measurable Payoffs'. In: *Dynamic Games and Applications* 5.1 (2015), pp. 120–135. DOI: `10.1007/s13235-014-0122-2` (cited on page 58).

[McN66]   Robert McNaughton. 'Testing and Generating Infinite Sequences by a Finite Automaton'. In: *Information and Control* 9.5 (1966), pp. 521–530. DOI: `10.1016/S0019-9958(66)80013-X` (cited on pages 5, 40).

[McN93]   Robert McNaughton. 'Infinite Games Played on Finite Graphs'. In: *Annals of Pure and Applied Logic* 65.2 (1993), pp. 149–184. ᴅᴏɪ: `10.1016/0168-0072(93)90036-D` (cited on pages 5, 23, 39, 100, 151).

[Mea55]   George H. Mealy. 'A method for synthesizing sequential circuits'. In: *The Bell System Technical Journal* 34.5 (1955), pp. 1045–1079. ᴅᴏɪ: `10.1002/j.1538-7305.1955.tb03788.x` (cited on pages 7, 25).

[Mey75]   Albert R. Meyer. 'Weak monadic second order theory of succesor is not elementary-recursive'. In: *Lecture Notes in Mathematics* 453 (1975), pp. 132–154. ᴅᴏɪ: `10.1007/BFb0064872` (cited on page 5).

[MM21]   Richard Mayr and Eric Munday. 'Strategy Complexity of Mean Payoff, Total Payoff and Point Payoff Objectives in Countable MDPs'. In: *Proceedings of the 32nd International Conference on Concurrency Theory, CONCUR 2021, Virtual Conference, August 24–27, 2021*. Ed. by Serge Haddad and Daniele Varacca. Vol. 203. LIPIcs. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021, 12:1–12:15. ᴅᴏɪ: `10.4230/LIPIcs.CONCUR.2021.12` (cited on page 240).

[Mos84]   Andrzej W. Mostowski. 'Regular expressions for infinite trees and a standard form of automata'. In: *Proceedings of the 5th Symposium on Computation Theory, SCT 1984, Zaborów, Poland, December 3–8, 1984*. Ed. by Andrzej Skowron. Vol. 208. Lecture Notes in Computer Science. Springer, 1984, pp. 157–168. ᴅᴏɪ: `10.1007/3-540-16066-3_15` (cited on pages 101, 109).

[Mos91]   Andrzej W. Mostowski. 'Games with Forbidden Positions'. In: Uniwersytet Gdański. Instytut Matematyki 78 (1991) (cited on pages 7, 11, 29, 109, 111).

[MP95]   Oded Maler and Amir Pnueli. 'On the Learnability of Infinitary Regular Sets'. In: *Information and Computation* 118.2 (1995), pp. 316–326. ᴅᴏɪ: `10.1006/inco.1995.1070` (cited on page 111).

[MPR20]   Benjamin Monmege, Julie Parreaux, and Pierre-Alain Reynier. 'Reaching Your Goal Optimally by Playing at Random with No Memory'. In: *Proceedings of the 31st International Conference on Concurrency Theory, CONCUR 2020, Vienna, Austria, September 1–4, 2020*. Ed. by Igor Konnov and Laura Kovács. Vol. 171. LIPIcs. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020, 26:1–26:21. ᴅᴏɪ: `10.4230/LIPIcs.CONCUR.2020.26` (cited on pages 98, 239).

[MR22]   James C. A. Main and Mickael Randour. 'Different Strokes in Randomised Strategies: Revisiting Kuhn's Theorem Under Finite-Memory Assumptions'. In: *Proceedings of the 33rd International Conference on Concurrency Theory, CONCUR 2022, Warsaw, Poland, September 12–16, 2022*. Ed. by Bartek Klin, Sławomir Lasota, and Anca Muscholl. Vol. 243. LIPIcs. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022, 22:1–22:18. ᴅᴏɪ: `10.4230/LIPIcs.CONCUR.2022.22` (cited on page 239).

[MS97]   Oded Maler and Ludwig Staiger. 'On Syntactic Congruences for Omega-Languages'. In: *Theoretical Computer Science* 183.1 (1997), pp. 93–112. ᴅᴏɪ: `10.1016/S0304-3975(96)00312-X` (cited on pages 44, 101, 111, 196).

[MSTW21] Richard Mayr, Sven Schewe, Patrick Totzke, and Dominik Wojtczak. 'Simple Stochastic Games with Almost-Sure Energy-Parity Objectives are in NP and coNP'. In: *Proceedings of the 24th International Conference on Foundations of Software Science and Computation Structures, FoSSaCS 2021, Held as Part of ETAPS 2021, Luxembourg City, Luxembourg,*
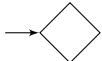
March 27 – April 1, 2021. Ed. by Stefan Kiefer and Christine Tasson. Vol. 12650. Lecture Notes in Computer Science. Springer, 2021, pp. 427–447. DOI: `10.1007/978-3-030-71995-1_22` (cited on pages 58, 238).

[Nas51]   John F. Nash. 'Non-cooperative Games'. In: *Annals of Mathematics* 54.2 (1951), pp. 286–295 (cited on pages 60, 240).
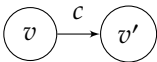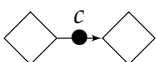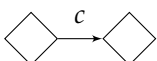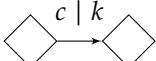
[Ner58]   A. Nerode. 'Linear Automaton Transformations'. In: *Proceedings of the American Mathematical Society* 9.4 (1958), pp. 541–544. DOI: `10.2307/2033204` (cited on pages 11, 43, 44, 101, 158, 203, 235).

[Ohl21]   Pierre Ohlmann. 'Monotonic graphs for parity and mean-payoff games'. PhD thesis. IRIF – Research Institute on the Foundations of Computer Science, 2021 (cited on pages 62, 195, 220, 221).

[Ohl23]   Pierre Ohlmann. 'Characterizing Positionality in Games of Infinite Duration over Infinite Graphs'. In: *TheoretiCS* 2 (2023), pp. 1–51. DOI: `10.46298/theoretics.23.3` (cited on pages 8, 13, 23, 96, 116, 194–196, 219–221, 226).

[OR94]   Martin J. Osborne and Ariel Rubinstein. *A course in game theory*. The MIT Press, 1994 (cited on pages 60, 62).

[Orn69]   Donald Ornstein. 'On the Existence of Stationary Optimal Strategies'. In: *Proceedings of the American Mathematical Society* 20.2 (1969), pp. 563–569 (cited on page 112).

[Pap94]   Christos H. Papadimitriou. *Computational complexity*. Addison-Wesley, 1994 (cited on page 155).

[Pnu77]   Amir Pnueli. 'The Temporal Logic of Programs'. In: *Proceedings of the 18th Annual Symposium on Foundations of Computer Science, FOCS 1977, Providence, RI, USA, October 31 – November 1, 1977.* IEEE Computer Society, 1977, pp. 46–57. DOI: `10.1109/SFCS.1977.32` (cited on pages 5, 24).

[PP04]   Dominique Perrin and Jean-Eric Pin. *Infinite words – automata, semigroups, logic and games*. Vol. 141. Pure and applied mathematics series. Elsevier Morgan Kaufmann, 2004 (cited on pages 65, 194).

[PR89]   Amir Pnueli and Roni Rosner. 'On the Synthesis of a Reactive Module'. In: *Proceedings of the 16th Annual ACM Symposium on Principles of Programming Languages, POPL 1989, Austin, TX, USA, January 11–13, 1989.* ACM Press, 1989, pp. 179–190. DOI: `10.1145/75277.75293` (cited on page 5).

[Put94]   Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley Series in Probability and Statistics. Wiley, 1994 (cited on pages 11, 29, 52).

[Rab69]   Michael O. Rabin. 'Decidability of Second-Order Theories and Automata on Infinite Trees'. In: *Transactions of the American Mathematical Society* 141 (1969), pp. 1–35. DOI: `10.2307/1995086` (cited on pages 5, 7, 39).

[Rén57]   Alfréd Rényi. 'Representations for real numbers and their ergodic properties'. In: *Acta Mathematica Academiae Scientiarum Hungarica* 8 (1957), pp. 477–493. DOI: `10.1007/BF02020331` (cited on pages 142, 143).

[RS59]   Michael O. Rabin and Dana S. Scott. 'Finite Automata and Their Decision Problems'. In: *IBM Journal of Research and Development* 3.2 (1959), pp. 114–125. DOI: `10.1147/rd.32.0114` (cited on page 183).

[Sha53]   Lloyd S. Shapley. 'Stochastic Games'. In: *Proceedings of the National Academy of Sciences* 39.10 (1953), pp. 1095–1100. DOI: `10.1073/pnas.39.10.1095` (cited on pages 8, 97, 112).

[Sta83]   Ludwig Staiger. 'Finite-State ω-Languages'. In: *Journal of Computer and System Sciences* 27.3 (1983), pp. 434–448. DOI: `10.1016/0022-0000(83)90051-X` (cited on pages 44, 101, 111, 196, 229).

[Tar72]   Robert E. Tarjan. 'Depth-First Search and Linear Graph Algorithms'. In: *SIAM Journal on Computing* 1.2 (1972), pp. 146–160. DOI: `10.1137/0201010` (cited on page 199).

[Tho08]   Wolfgang Thomas. 'Church's Problem and a Tour through Automata Theory'. In: *Pillars of Computer Science, Essays Dedicated to Boris (Boaz) Trakhtenbrot on the Occasion of His 85th Birthday*. Ed. by Arnon Avron, Nachum Dershowitz, and Alexander Rabinovich. Vol. 4800. Lecture Notes in Computer Science. Springer, 2008, pp. 635–655. DOI: `10.1007/978-3-540-78127-1_35` (cited on page 195).

[Tse70]   Grigori Tseitin. 'On the complexity of derivation in propositional calculus'. In: *Studies in Constructive Mathematics and Mathematical Logic: Part II*. Seminars in mathematics (1970), pp. 115–125. DOI: `10.1007/978-3-642-81955-1_28` (cited on page 192).

[Tur37]   Alan M. Turing. 'On Computable Numbers, with an Application to the Entscheidungsproblem'. In: *Proceedings of the London Mathematical Society* 2.42 (1 1937), pp. 230–265. DOI: `10.1112/plms/s2-42.1.230` (cited on page 2).

[VCD+15]   Yaron Velner, Krishnendu Chatterjee, Laurent Doyen, Thomas A. Henzinger, Alexander Moshe Rabinovich, and Jean-François Raskin. 'The complexity of multi-mean-payoff and multi-energy games'. In: *Information and Computation* 241 (2015), pp. 177–196. DOI: `10.1016/j.ic.2015.03.001` (cited on pages 8, 33, 57, 99).

[von28]   John von Neumann. 'Zur Theorie der Gesellschaftsspiele'. In: *Mathematische Annalen* 100 (1928), pp. 295–320 (cited on page 240).

[Wag79]   Klaus W. Wagner. 'On ω-Regular Sets'. In: *Information and Control* 43.2 (1979), pp. 123–177. DOI: `10.1016/S0019-9958(79)90653-3` (cited on pages 38, 229).

[Wal01]   Igor Walukiewicz. 'Pushdown Processes: Games and Model-Checking'. In: *Information and Computation* 164.2 (2001), pp. 234–263. DOI: `10.1006/inco.2000.2894` (cited on page 241).

[Zie98]   Wiesław 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. DOI: `10.1016/S0304-3975(98)00009-7` (cited on pages 11, 39, 101, 109, 221).

Table of notations

Graphical conventions

◯	arena vertex controlled by $\mathcal{P}_1$; in Chapters 7 and 8, also the vertex of a graph	⤳ w	sequence of edges or transitions labeled with word w
▢	arena vertex controlled by $\mathcal{P}_2$	→◇	initial memory or automaton state
◇	automaton or memory state	◈	final state of a DFA
v →c→ v'	edge (v, c, v') of an arena	◇—•—◇ c	Büchi transition of a DBA
◇ →c→ ◇	transition of automaton or memory structure with color c	◇ →$c \mid k$→ ◇	transition of a DPA with color c and priority k

Common mathematical symbols

$\mathbb{N}$	set $\{0, 1, \ldots\}$ of natural numbers
$\mathbb{Z}$	set of integers
$\mathbb{Q}$	set of rational numbers
$\mathbb{R}$	set of real numbers
$\emptyset$	empty set
A^*	set of finite sequences of elements of A
A^+	set of non-empty finite sequences of elements of A
A^ω	set of infinite sequences of elements of A
$\|A\|$	cardinality of A
ε	empty word
w, x, y	finite or infinite words
$\uplus$	disjoint union
$\preceq$	preorder
Γ	chain for a preorder
$\sim$	equivalence relation
$[a]_\sim$	equivalence class of a for equivalence relation $\sim$
$A/_\sim$	quotient of A by equivalence relation $\sim$
$\lceil \cdot \rceil$	ceiling function
$\lfloor \cdot \rfloor$	floor function
$\mathcal{O}(\cdot)$	big O notation

Arenas

C	non-empty set of colors
$a, b, c \in C$	colors
$\mathcal{P}_1, \mathcal{P}_2$	Player 1, Player 2
ℓ	index of a player (in $\{1, 2\}$)

$\mathcal{A} = (V, V_1, V_2, E)$	game arena
V	set of vertices
V_1, V_2	set of vertices controlled resp. by $\mathcal{P}_1$, by $\mathcal{P}_2$
$u, v \in V$	vertices
$E \subseteq V \times C \times V$	set of colored edges
$e = (\mathsf{in}(e), \mathsf{col}(e), \mathsf{out}(e)) \in E$	colored edge
$\gamma = e_1 \ldots e_n \in E^+$	non-empty history of an arena $\mathcal{A} = (V, V_1, V_2, E)$
λ_v	empty history from $v \in V$
$\mathsf{col}^*(\gamma) = \mathsf{col}(e_1) \ldots \mathsf{col}(e_n) \in C^*$	projection of a history to colors
$\mathsf{Hists}(\mathcal{A})$	set of histories of arena $\mathcal{A}$
$\mathsf{Hists}_\ell(\mathcal{A})$	set of histories of $\mathcal{A}$ ending in a vertex controlled by $\mathcal{P}_\ell$
$\rho = e_1 e_2 \ldots \in E^\omega$	play of an arena $\mathcal{A} = (V, V_1, V_2, E)$
$\mathsf{col}^\omega(\rho) = \mathsf{col}(e_1) \mathsf{col}(e_2) \ldots \in C^\omega$	projection of a play to colors
$\mathsf{Plays}(\mathcal{A})$	set of plays of arena $\mathcal{A}$
$\sigma_\ell \colon \mathsf{Hists}_\ell(\mathcal{A}) \to E$	strategy of $\mathcal{P}_\ell$ on $\mathcal{A} = (V, V_1, V_2, E)$

Objectives and games

$W \subseteq C^\omega$	objective
$\overline{W}$	complement $C^\omega \setminus W$ of objective W
$\mathcal{G} = (\mathcal{A}, W)$	game
$\mathsf{Reach}(a)$	reachability condition with target color $a \in C$
$\mathsf{Safe}(a)$	safety condition with color $a \in C$ to avoid
$\mathsf{Büchi}(a)$	Büchi condition with color $a \in C$ to see infinitely often
$\mathsf{Parity}(n)$	parity condition on colors $C = \{0, \ldots, n\}$
$\mathsf{Muller}(\mathcal{F})$	Muller condition on $\mathcal{F} \subseteq 2^C$
$\mathsf{MP}^{\geq 0}$	set of infinite words of rational numbers with non-negative mean payoff
$w^{-1} W$	winning continuations of $w \in C^*$ for W
$\preceq_W$	prefix preorder of W (subscript W often omitted)
$\prec_W$	strict prefix preorder of W (subscript W often omitted)
$\sim_W$	right congruence of W (subscript W often omitted)

Memory structures

$\mathcal{M} = (M, m_{\mathsf{init}}, \alpha_{\mathsf{upd}})$	memory structure on C
M	set of memory states
$m \in M$	memory state
$m_{\mathsf{init}} \in M$	initial memory state
$\alpha_{\mathsf{upd}} \colon M \times C \to M$	memory update function
$\alpha_{\mathsf{upd}}^* \colon M \times C^* \to M$	extended memory update function
$\mathcal{M}_{\mathsf{triv}}$	trivial memory structure (with $M = \{m_{\mathsf{init}}\}$)
$\mathcal{M}_1 \otimes \mathcal{M}_2$	direct product of $\mathcal{M}_1$ and $\mathcal{M}_2$
L_{m_1, m_2}	finite words read from m_1 to m_2 in $\mathcal{M}$
$\alpha_{\mathsf{nxt}} \colon V_\ell \times M \to E$	next-action function of $\mathcal{P}_\ell$

Automata

$\mathcal{S} = (Q, \Sigma, q_{\text{init}}, \delta)$	automaton structure
$\mathcal{D} = (Q, \Sigma, q_{\text{init}}, \delta, F)$	deterministic finite automaton (DFA) (can be infinite in a few clearly labeled occurrences in Chapter 7)
$\mathcal{B} = (Q, \Sigma, q_{\text{init}}, \delta, B)$	deterministic Büchi automaton (DBA)
$\mathcal{P} = (Q, \Sigma, q_{\text{init}}, \delta, p)$	deterministic parity automaton (DPA)
Q	set of automaton states
$q \in Q$	automaton state
Σ	alphabet (often, $\Sigma = C$)
$q_{\text{init}} \in Q$	initial automaton state
$\delta \colon Q \times C \to Q$	complete, deterministic update function
$F \subseteq Q$	set of final states
$B \subseteq Q \times C$	set of Büchi transitions
$p \colon M \times C \to \{0, \dots, n\}$	transition-based priority function
$\mathcal{L}(\cdot)$	language (of finite or infinite words) recognized by an automaton
ϱ	finite or infinite run of an automaton
$\mathcal{S}_W = (Q_W, C, q_{\text{init}}^W, \delta_W)$	prefix classifier of W

Notations specific to Chapter 4

$\sqsubseteq \; \subseteq C^\omega \times C^\omega$	preference relation
$\mathcal{G} = (\mathcal{A}, \sqsubseteq)$	quantitative game
$\sqsubseteq^{-1}$	inverse of a preference relation
$\sqsubset$	strict preference relation
$\mathsf{UCol}_\sqsubseteq(\mathcal{A}, v, \sigma_1)$	upward closure for $\sqsubseteq$ of plays consistent with strategy σ_1 of $\mathcal{P}_1$
$\mathsf{DCol}_\sqsubseteq(\mathcal{A}, v, \sigma_2)$	downward closure for $\sqsubseteq$ of plays consistent with strategy σ_2 of $\mathcal{P}_2$
$\mathcal{A} \ltimes \mathcal{M}$	arena resulting from the product of arena $\mathcal{A}$ and memory structure $\mathcal{M}$
$\mathcal{N} = (Q, C, \Delta, Q_{\text{init}}, F)$	non-deterministic finite automaton
$\Delta \subseteq Q \times C \times Q$	transitions of a non-deterministic automaton
$Q_{\text{init}} \subseteq Q$	set of initial states of a non-deterministic automaton
$K \subseteq C^*$	set of finite words
$[K]$	set of infinite words whose prefixes are all prefixes of words in K
V_{cov}	set of vertices of an arena prefix-covered and/or cyclic-covered by a memory structure
$\mathfrak{A}$	set of arenas

Notations specific to Chapter 5

$\pi \in (M \times C)^+$	non-empty path of memory structure $\mathcal{M} = (M, m_{\text{init}}, \alpha_{\text{upd}})$
$(m, \_)$	empty path from $m \in M$ of memory structure $\mathcal{M}$
$\mathsf{col}^*(\pi) \in C^*$	projection of path π to its sequence of colors

$\mathrm{st}(\pi) \subseteq M$	set of memory states visited by π
"$(m, c) \in \pi$"	transition $(m, c) \in M \times C$ is part of path π
$\varphi, \psi \in (M \times C)^+$	cycles of memory structure $\mathcal{M}$
Π_{m_1, m_2}	set of paths from m_1 to m_2, with $m_1, m_2 \in M$
Φ_m	cycles on $m \in M$
$\Phi_{\mathcal{M}}$	all cycles of $\mathcal{M}$
$\Phi_{\mathcal{M}}^{\mathsf{win}, w}$	cycles on the memory state reached by w in $\mathcal{M}$ that induce winning words when repeated after w, and ...
$\Phi_{\mathcal{M}}^{\mathsf{lose}, w}$	... their losing counterparts
$\Phi_m^{\mathsf{win}}, \Phi_m^{\mathsf{lose}}$	winning, losing cycles on $\mathcal{M}$ (under hypotheses on $\mathcal{M}$)
$\mathsf{val}(\varphi)$	value of a cycle (win or lose) (under hypotheses on $\mathcal{M}$)
$\overline{\varphi}$	cycle witnessing the competition of two cycles
$\mathsf{comp}(\varphi), \mathsf{domBy}(\varphi)$	cycles competing with φ, cycles dominated by φ
$\lhd, \simeq$	preorder on cycles, equivalence relation on cycles

Notations specific to Chapter 7

$\mathsf{GenReach}(A)$	general reachability objective derived from $A \subseteq C^*$
$\mathsf{GenSafe}(A)$	general safety objective derived from $A \subseteq C^*$
$\Gamma_m^{\mathcal{D}}$	set of states of $\mathcal{D}$ reachable using a finite word also reaching memory state m
$\mathcal{D}_W$	prefix-classifier automaton of a general reachability or safety objective W
θ	ordinal
$\mathsf{rank}(\gamma)$	rank of node γ in a well-founded tree
$\mathcal{A}^{\sigma, v}$	tree induced by strategy σ on $\mathcal{A}$ from v
$\mathcal{A}^{\sigma, v}_{|\mathcal{L}(\mathcal{D})}$	same, but with branches cut below nodes in $\mathcal{L}(\mathcal{D})$
$G = (V, E)$	graph
$E \subseteq V \times V$	non-colored edges
$\mathcal{C}_n = (V_{\mathcal{C}}, E_{\mathcal{C}})$	cycle graph with n vertices

Notations specific to Chapter 8

$\mathsf{Büchi}(C')$	Büchi condition with colors in $C' \subseteq C$ to see infinitely often
$\mathcal{B}(q, w)$	finite or infinite run of DBA $\mathcal{B}$ on w from q
$B\text{-}\mathsf{Free}_{\mathcal{B}}(q)$	B-free words of $\mathcal{B}$ from q
$B\text{-}\mathsf{FreeCycles}_{\mathcal{B}}(q)$	B-free cycles of $\mathcal{B}$ on q
θ, λ, η	ordinals
κ	cardinal
$\mathcal{U}$	universal graph
$v \xrightarrow{c} v'$	edge (v, c, v') of a graph
ϕ	graph morphism