Arena-Independent Memory Bounds for Nash Equilibria in Reachability Games

James C. A. Main
F.R.S.-FNRS & UMONS – Université de Mons, Belgium

Abstract

We study the memory requirements of Nash equilibria in turn-based multiplayer games on possibly *infinite graphs* with reachability, shortest path and Büchi objectives.

We present constructions for *finite-memory* Nash equilibria in these games that apply to arbitrary game graphs, bypassing the finite-arena requirement that is central in existing approaches. We show that, for these three types of games, from any Nash equilibrium, we can derive another Nash equilibrium where all strategies are finite-memory such that the same players accomplish their objective, without increasing their cost for shortest path games.

Furthermore, we provide memory bounds that are *independent of the size of the game graph* for reachability and shortest path games. These bounds depend only on the number of players.

To the best of our knowledge, we provide the first results pertaining to finite-memory constrained Nash equilibria in infinite arenas and the first arena-independent memory bounds for Nash equilibria.

2012 ACM Subject Classification Theory of computation → Solution concepts in game theory

Keywords and phrases multiplayer games on graphs, Nash equilibrium, finite-memory strategies

Digital Object Identifier 10.4230/LIPIcs.STACS.2024.50

Related Version *Full Version*: https://arxiv.org/abs/2310.02142 [20]

Funding This work has been supported by the Fonds de la Recherche Scientifique - FNRS under Grant n° T.0188.23 (PDR ControlleRS). The author is a Research Fellow of the Fonds de la Recherche Scientifique – FNRS and member of the TRAIL institute.

Acknowledgements I thank Thomas Brihaye, Aline Goeminne and Mickael Randour for fruitful discussions and their comments on a preliminary version of this paper.

1 Introduction

Games on graphs. *Games on graphs* are a prevalent framework to model reactive systems, i.e., systems that continuously interact with their environment. Typically, this interaction is modelled as an infinite-duration *two-player (turn-based) zero-sum game* played on an arena (i.e., a game graph) where a system player and an environment player are adversaries competing for opposing goals (e.g., [18, 1, 14]), which can be modelled, e.g., by numerical costs for the system player. Determining whether the system can enforce some specification boils down to computing how low of a cost the system player can guarantee. We then construct an *optimal strategy* for the system which can be seen as a *formal blueprint* for a controller of the system to be implemented [24, 1]. For implementation purposes, strategies should have a finite representation. We consider *finite-memory strategies* (e.g., [3]) which are strategies defined by Mealy machines, i.e., automata with outputs on their edges.

Nash equilibria. In some applications, this purely adversarial model may be too restrictive. This is the case in settings with several agents, each with their own objective, who are not necessarily opposed to one another. Such situations are modelled by *multiplayer non-zero-sum games* on graphs. The counterpart of optimal strategies in this setting is typically a notion

41st International Symposium on Theoretical Aspects of Computer Science (STACS 2024).
Editors: Olaf Beyersdorff, Mamadou Moustapha Kanté, Orna Kupferman, and Daniel Lokshtanov; Article No. 50; pp. 50:1–50:18
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

of *equilibrium*. We focus on *Nash equilibria* [22] (NEs) in the following; an NE is a tuple of strategies, one per player, such that no player has an incentive to unilaterally deviate from their strategy.

Reachability games. We focus on variants of *reachability* games on possibly infinite arenas. In a reachability game, the goal of each player is given by a set of target vertices to be visited. We also study *shortest path* games, where players aim to visit their targets as soon as possible (where time is modelled by non-negative edge weights), and *Büchi* games, where players aim to visit their targets infinitely often. NEs are guaranteed to exist for these games: see [7, 12] for reachability games and for shortest path games in finite arenas, the full version of this work [20] for shortest path games in general and [25] for Büchi games.

Usually, finite-memory NEs for these games are given by strategies whose size depends on the arena (e.g., [7, 26, 6]). These constructions consequently do not generalise to infinite arenas. The main idea of these approaches is as follows. First, one shows that there exist plays resulting from NEs with a *finite representation*, e.g., a lasso. This play is then encoded in a Mealy machine. If some player is inconsistent with the play, the other players switch to a (finite-memory) *punishing strategy* to sabotage the deviating player; this enforces the stability of the equilibrium. This punishing mechanism is inspired by the proof of the folk theorem for NEs in repeated games [15, 23].

Contributions. Our contributions are twofold. First, we present constructions for *finite-memory* NEs for reachability, shortest path and Büchi games that apply to arbitrary arenas, bypassing the finite-arena requirement that is central in existing approaches. More precisely, for these three types of games, we show that from any NE, we can derive another NE where all strategies are finite-memory and such that the same players accomplish their objective, without increasing their cost for shortest path games. In other words, our constructions are general and can be used to match or improve any NE cost profile.

Second, for reachability and shortest-path games, we provide memory bounds that are *independent* of the size of the arena which are quadratic in the number of players.

Our key observation is that it is not necessary to fully implement the punishment mechanism: some deviations do not warrant switching to punishing strategies. This allows us to encode only part of the information in the memory instead of an entire play.

Related work. We refer to the survey [9] for an extensive bibliography on games played on finite graphs, to [8] for a survey centred around reachability games and to [14] as a general reference on games on graphs. We discuss three research direction related to this work.

The first direction is related to *computational problems* for NEs. In the settings we consider, NEs are guaranteed to exist. However, NEs where no player satisfy their objective can coexist with NEs where all players satisfy their objective [25, 26]. A classical problem is to decide if there exists a constrained NE, i.e., such that certain players satisfy their objective in the qualitative case or such that the cost incurred by players is bounded from above in a quantitative case (e.g., [10, 2]). Deciding the existence of a constrained NE is NP-complete for reachability and shortest path games [6] and is in P for Büchi objectives [26].

Second, the construction of our finite-memory NEs rely on *characterisations* of plays resulting from NEs. Their purpose is to ensure that the punishment mechanism described above can be used to guarantee the stability of an equilibrium. In general, these characterisations can be useful from an algorithmic perspective; deciding the existence of a constrained NE boils down to finding a play that satisfies the characterisation. Characterisations appear in the literature for NEs [26, 27, 2], but also for other types of equilibria, e.g., subgame perfect equilibria [5] and secure equilibria [10].

Finally, there exists a body of work dedicated to better understanding the complexity of optimal strategies in zero-sum games. We mention [17] for memoryless strategies, and [3] and [4] for finite-memory strategies in finite and infinite arenas respectively. In finite arenas, for the finite-memory case, a key notion is *arena-independent* finite-memory strategies, i.e., strategies based on a memory structure that is sufficient to win in all arenas whenever possible. In this work, the finite-memory strategies we propose actually depend on the arena; only their size does not. We also mention [19]: in games on finite arenas with objectives from a given class, finite-memory NEs exist if certain conditions on the corresponding zero-sum games hold.

Outline. Due to space constraints, we only provide an overview of our work: technical details can be found in the full paper [20]. This work is structured as follows. In Sect. 2, we summarise prerequisite definitions. We establish the existence of memoryless punishing strategies by studying zero-sum games in Sect. 3. Characterisations of NEs are provided in Sect. 4. We prove our main results on finite-memory NEs in reachability and shortest path games in Sect. 5. Finally, Sect. 6 is dedicated to the corresponding result for Büchi games.

2 Preliminaries

Notation. We write $\mathbb{N}$, $\mathbb{R}$ for the sets of natural and real numbers respectively, and let $\overline{\mathbb{R}} = \mathbb{R} \cup \{+\infty, -\infty\}$ and $\overline{\mathbb{N}} = \mathbb{N} \cup \{+\infty\}$. For any $n \in \mathbb{N}$, $n \geq 1$, we let $[\![n]\!] = \{1, \dots, n\}$.

Games. Let (V, E) be a directed graph where V is a (possibly infinite) set of vertices and $E \subseteq V \times V$ is an edge relation. For any $v \in V$, we write $\mathsf{Succ}_E(v) = \{v' \in V \mid (v, v') \in E\}$ for the set of successor vertices of v. An *n-player arena* is a tuple $\mathcal{A} = ((V_i)_{i\in[\![n]\!]}, E)$, where $(V_i)_{i\in[\![n]\!]}$ is a partition of V. We assume that there are no deadlocks in the arenas we consider, i.e., for all $v \in V$, $\mathsf{Succ}_E(v)$ is not empty. We write $\mathcal{P}_i$ for player i.

A play starts in an initial vertex and proceeds as follows. At each round of the game, the player controlling the current vertex selects a successor of this vertex and the current vertex is updated accordingly. The play continues in this manner infinitely. Formally, a *play* of $\mathcal{A}$ is an infinite sequence $v_0v_1 \ldots \in V^\omega$ such that $(v_\ell, v_{\ell+1}) \in E$ for all $\ell \in \mathbb{N}$. For a play $\pi = v_0v_1 \ldots$ and $\ell \in \mathbb{N}$, we let $\pi_{\geq \ell} = v_\ell v_{\ell+1} \ldots$ denote the suffix of π from position ℓ and $\pi_{\leq \ell} = v_0 \ldots v_\ell$ denote the prefix of π up to position ℓ. A *history* is any finite non-empty prefix of a play. We write $\mathsf{Plays}(\mathcal{A})$ and $\mathsf{Hist}(\mathcal{A})$ for the set of plays and histories of $\mathcal{A}$ respectively. For $i \in [\![n]\!]$, we let $\mathsf{Hist}_i(\mathcal{A}) = \mathsf{Hist}(\mathcal{A}) \cap V^*V_i$. For any history $h = v_0 \ldots v_r$, we let $\mathsf{first}(h)$ and $\mathsf{last}(h)$ respectively denote v_0 and v_r. For any play π, $\mathsf{first}(\pi)$ is defined similarly.

We formalise the goal of a player in two ways. In the qualitative case, we describe the goal of a player by a set of plays, called an *objective*. We say that a play π satisfies an objective Ω if $\pi \in \Omega$. For quantitative specifications, we assign to each play a quantity using a *cost function* $\mathsf{cost}_i \colon \mathsf{Plays}(\mathcal{A}) \to \overline{\mathbb{R}}$ that $\mathcal{P}_i$ intends to minimise. Any goal expressed by an objective Ω can be encoded using a cost function cost_i which assigns 0 to plays in Ω and 1 to others; aiming to minimise this cost is equivalent to aiming to satisfy the objective. For this reason, we present further definitions using cost functions, and explicitly mention when notions are specific to objectives.

A *game* is an arena augmented with the goals of each player. Formally, a game is a tuple $\mathcal{G} = (\mathcal{A}, (\mathsf{cost}_i)_{i\in[\![n]\!]})$ where $\mathcal{A}$ is an arena and, for all $i \in [\![n]\!]$, cost_i is the cost function of $\mathcal{P}_i$. The *cost profile* of a play π is $(\mathsf{cost}_i(\pi))_{i\in[\![n]\!]}$. Given two plays π and π', we say that the cost profile of π is preferable to that of π' if $\mathsf{cost}_i(\pi) \leq \mathsf{cost}_i(\pi')$ for all $i \in [\![n]\!]$.

Objectives and costs. We consider a qualitative and quantitative formulation for the goal of reaching a target, and the goal of infinitely often reaching a target. Let $T \subseteq V$ denote a set of target vertices. We often refer to the set T as a *target*.

We first consider the reachability objective, which expresses the goal of reaching T. Formally, the *reachability objective* (for T) $\mathsf{Reach}(T)$ is defined by the set $\{v_0v_1v_2\ldots \in \mathsf{Plays}(\mathcal{A}) \mid \exists\, \ell \in \mathbb{N},\, v_\ell \in T\}$. The complement of the reachability objective $\mathsf{Safe}(T) = \mathsf{Plays}(\mathcal{A}) \setminus \mathsf{Reach}(T)$, which expresses the goal of avoiding T, is called the *safety* objective.

Second, we introduce a cost function formalising the goal of reaching a target as soon as possible. In this context, we assign (non-negative) weights to edges via a *weight function* $w\colon E \to \mathbb{N}$, which model, e.g., the time taken when traversing an edge. The weight function is extended to histories as follows; for $h = v_0 \ldots v_r \in \mathsf{Hist}(\mathcal{A})$, we let $w(h) = \sum_{\ell=0}^{r-1} w((v_\ell, v_{\ell+1}))$. We define the *truncated sum* cost function (for T and w), for all plays $\pi = v_0v_1\ldots$, by $\mathsf{TS}_w^T(\pi) = w(\pi_{\leq r})$ if $r = \min\{\ell \in \mathbb{N} \mid v_\ell \in T\}$ exists and $\mathsf{TS}_w^T(\pi) = +\infty$ otherwise.

Finally, we define the Büchi objective, expressing the goal of reaching a target infinitely often. Formally, the *Büchi objective* (for T) $\mathsf{Büchi}(T)$ is defined by the set $\{v_0v_1v_2\ldots \in \mathsf{Plays}(\mathcal{A}) \mid \forall\, \ell \in \mathbb{N},\, \exists\, \ell' \geq \ell,\, v_{\ell'} \in T\}$. The complement of a Büchi objective is a co-Büchi objective: the *co-Büchi* objective (for T), which expresses the goal of visiting T finitely often, is defined as $\mathsf{coBüchi}(T) = \mathsf{Plays}(\mathcal{A}) \setminus \mathsf{Büchi}(T)$.

We refer to games where all players have a reachability objective (resp. a truncated sum cost function, a Büchi objective) as *reachability* (resp. *shortest path*, *Büchi*) *games*.

Let $T_1, \ldots, T_n \subseteq V$ be targets for each player and $\pi = v_0v_1\ldots \in \mathsf{Plays}(\mathcal{A})$. For reachability and shortest path games, we introduce the notation $\mathsf{VisPl}_{T_1,\ldots,T_n}(\pi) = \{i \in [\![n]\!] \mid \pi \in \mathsf{Reach}(T_i)\}$ as the set of players whose targets are visited in π and $\mathsf{VisPos}_{T_1,\ldots,T_n}(\pi) = \{\min\{\ell \in \mathbb{N} \mid v_\ell \in T_i\} \mid i \in \mathsf{VisPl}(\pi)\}$ as the set of earliest positions at which targets are visited along π. For Büchi games, we define $\mathsf{InfPl}_{T_1,\ldots,T_n}(\pi) = \{i \in [\![n]\!] \mid \pi \in \mathsf{Büchi}(T_i)\}$ as the set of players whose target is visited infinitely often in π. When $T_1, \ldots, T_n$ are clear from the context, we omit them.

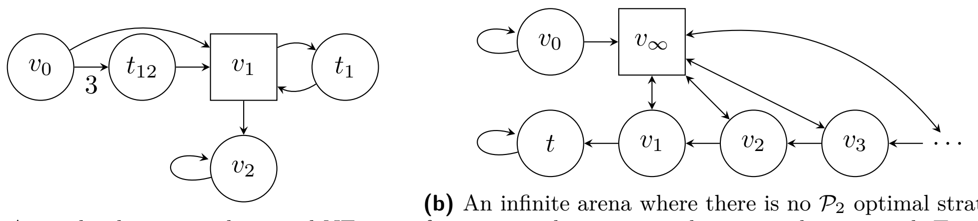
Strategies. Strategies describe the decisions of players during a play. These choices may depend on the past, and not only the current vertex of the play. Formally, a *strategy* of $\mathcal{P}_i$ in an arena $\mathcal{A}$ is a function $\sigma_i\colon \mathsf{Hist}_i(\mathcal{A}) \to V$ such that for all histories $h \in \mathsf{Hist}_i(\mathcal{A})$, $(\mathsf{last}(h), \sigma_i(h)) \in E$. A *strategy profile* is a tuple $\sigma = (\sigma_i)_{i\in[\![n]\!]}$, where σ_i is a strategy of $\mathcal{P}_i$ for all $i \in [\![n]\!]$. To highlight the role of $\mathcal{P}_i$, we sometimes write $\sigma = (\sigma_i, \sigma_{-i})$, where σ_{-i} denotes the strategy profile of the players other than $\mathcal{P}_i$.

A play $\pi = v_0v_1v_2\ldots$ is *consistent* with a strategy σ_i of $\mathcal{P}_i$ if for all $\ell \in \mathbb{N}$, $v_\ell \in V_i$ implies $v_{\ell+1} = \sigma_i(\pi_{\leq \ell})$. A play is *consistent* with a strategy profile if it is consistent with all strategies of the profile. Given an initial vertex v_0 and a strategy profile σ, there is a unique play $\mathsf{Out}(\sigma, v_0)$ from v_0 that is consistent with σ, called the *outcome* of σ from v_0.

We identify two classes of strategies of interest in this work. A strategy σ_i is *memoryless* if the moves it prescribes depend only on the current vertex, i.e., if for all h, $h' \in \mathsf{Hist}_i(\mathcal{A})$, if $\mathsf{last}(h) = \mathsf{last}(h')$, then $\sigma_i(h) = \sigma_i(h')$. We view memoryless strategies as functions $V_i \to V$.

A strategy is *finite-memory* if it can be encoded by a Mealy machine, i.e., a finite automaton with outputs. A *Mealy machine* (for $\mathcal{P}_i$) is a tuple $\mathcal{M} = (M, m_{\mathsf{init}}, \mathsf{up}, \mathsf{nxt}_i)$ where M is a finite set of memory states, m_{init} is an initial memory state, $\mathsf{up}\colon M \times V \to M$ is a memory update function and $\mathsf{nxt}_i\colon M \times V_i \to V$ is a next-move function.

To describe the strategy induced by a Mealy machine, we first define the iterated update function $\widehat{\mathsf{up}}\colon V^* \to M$ by induction. We write ε for the empty word. We let $\widehat{\mathsf{up}}(\varepsilon) = m_{\mathsf{init}}$ and for all $wv \in V^*$, $\widehat{\mathsf{up}}(wv) = \mathsf{up}(\widehat{\mathsf{up}}(w), v)$. The strategy $\sigma_i^{\mathcal{M}}$ induced by $\mathcal{M}$ is defined, for all histories $h = h'v \in \mathsf{Hist}_i(\mathcal{A})$, by $\sigma_i^{\mathcal{M}}(h) = \mathsf{nxt}_i(\widehat{\mathsf{up}}(h'), v)$.



(a) A weighted arena with several NEs.

(b) An infinite arena where there is no $\mathcal{P}_2$ optimal strategy from v_∞ in the zero-sum shortest path game with $T = \{t\}$.

Figure 1 Two weighted arenas. Circles and squares respectively denote $\mathcal{P}_1$ and $\mathcal{P}_2$ vertices. Edge labels denote their weight and unlabelled edges have a weight of 1.

We say that a finite-memory strategy σ_i has *memory size* $b \in \mathbb{N}$ if there is some Mealy machine $(M, m_{\mathsf{init}}, \mathsf{up}, \mathsf{nxt}_i)$ encoding σ_i with $|M| = b$ and b is the smallest such number.

▶ Remark 1. Some authors define the updates of Mealy machines using edges rather than vertices. Any vertex-update Mealy machine can directly be seen as an edge-update Mealy machine. The converse is not true. In particular, a vertex-update Mealy machine representation of a strategy can require a larger size than an equivalent edge-update Mealy machine.

Nash equilibria. Let $\mathcal{G} = (\mathcal{A}, (\mathsf{cost}_i)_{i\in[\![n]\!]})$ be a game and v_0 be an initial vertex. Given a strategy profile $\sigma = (\sigma_i)_{i\in[\![n]\!]}$, we say that a strategy τ_i of $\mathcal{P}_i$ is a *profitable deviation* (with respect to σ from v_0) if $\mathsf{cost}_i(\mathsf{Out}((\tau_i, \sigma_{-i}), v_0)) < \mathsf{cost}_i(\mathsf{Out}(\sigma, v_0))$. A *Nash equilibrium* (NE) from v_0 is a strategy profile such that no player has a profitable deviation. Equivalently, σ is an NE from v_0 if, for all $i \in [\![n]\!]$ and all plays π consistent with σ_{-i} starting in v_0, $\mathsf{cost}_i(\pi) \geq \mathsf{cost}_i(\mathsf{Out}(\sigma, v_0))$. In general, NEs with incomparable cost profiles may coexist.

▶ **Example 2.** Consider the shortest path game played on the arena depicted in Fig. 1a where $T_1 = \{t_{12}, t_1\}$ and $T_2 = \{t_{12}\}$. The memoryless strategy profile (σ_1, σ_2) with $\sigma_1(v_0) = t_{12}$ and $\sigma_2(v_1) = v_2$ is an NE from v_0 with cost profile $(3, 3)$. Another NE from v_0 would be the memoryless strategy profile (σ'_1, σ'_2) such that $\sigma'_1(v_0) = v_1$ and $\sigma'_2(v_1) = t_1$; the cost profile of its outcome is $(2, +\infty)$, which is incomparable with $(3, 3)$. ⌟

Zero-sum games. In a zero-sum game, two players compete with opposing goals. Formally, a two-player zero-sum game is a two-player game $\mathcal{G} = (\mathcal{A}, (\mathsf{cost}_1, \mathsf{cost}_2))$ where $\mathcal{A}$ is a two-player arena and $\mathsf{cost}_2 = -\mathsf{cost}_1$. We usually shorten the notation of a zero-sum game to $\mathcal{G} = (\mathcal{A}, \mathsf{cost}_1)$ due to the definition.

Let $v_0 \in V$. If $\inf_{\sigma_1} \sup_{\sigma_2} \mathsf{cost}_1(\mathsf{Out}((\sigma_1, \sigma_2), v_0)) = \sup_{\sigma_2} \inf_{\sigma_1} \mathsf{cost}_1(\mathsf{Out}((\sigma_1, \sigma_2), v_0))$, where σ_i is quantified over the strategies of $\mathcal{P}_i$, we refer to the above as the *value* of v_0 and denote it by $\mathsf{val}(v_0)$. A game is *determined* if the value is defined in all vertices.

A strategy σ_1 of $\mathcal{P}_1$ (resp. σ_2 of $\mathcal{P}_2$) is said to ensure $\alpha \in \overline{\mathbb{R}}$ from a vertex v_0 if all plays π consistent with σ_1 (resp. σ_2) from v_0 are such that $\mathsf{cost}_1(\pi) \leq \alpha$ (resp. $\mathsf{cost}_1(\pi) \geq \alpha$). A strategy of $\mathcal{P}_i$ is *optimal* from $v_0 \in V$ if it ensures $\mathsf{val}(v_0)$ from v_0. A strategy is a *uniform optimal* strategy if it ensures $\mathsf{val}(v)$ from v for all $v \in V$. Optimal strategies do not necessarily exist, even if the value does.

▶ **Example 3.** Consider the two-player zero-sum game played on the weighted arena illustrated in Fig. 1b where the cost function of $\mathcal{P}_1$ is $\mathsf{TS}_w^{\{t\}}$. Let $\alpha \in \mathbb{N} \setminus \{0\}$. It holds that $\mathsf{val}(v_\alpha) = \alpha$. On the one hand, $\mathcal{P}_1$ can ensure a cost of α from v_α by moving leftward in the illustration. On the other hand, $\mathcal{P}_2$ can ensure a cost of α from v_α with the memoryless strategy that moves from v_∞ to v_α. It follows that this same memoryless strategy of $\mathcal{P}_2$ ensures $\alpha + 1$ from v_∞. We conclude that $\mathsf{val}(v_\infty) = +\infty$. However, $\mathcal{P}_2$ cannot prevent t from being reached from v_∞, despite its infinite value. Therefore, $\mathcal{P}_2$ does not have an optimal strategy. ⌟

If the goal of $\mathcal{P}_1$ is formulated by an objective Ω, we say that a strategy σ_1 of $\mathcal{P}_1$ (resp. σ_2 of $\mathcal{P}_2$) is *winning* from v_0 if all plays consistent with it from v_0 satisfy Ω (resp. $\mathsf{Plays}(\mathcal{A}) \setminus \Omega$). The set of vertices from which $\mathcal{P}_1$ (resp. $\mathcal{P}_2$) has a winning strategy is called their *winning region* denoted by $W_1(\Omega)$ (resp. $W_2(\mathsf{Plays}(\mathcal{A}) \setminus \Omega)$). A strategy σ_1 of $\mathcal{P}_1$ (resp. σ_2 of $\mathcal{P}_2$) is a *uniform winning* strategy if it is winning from all vertices in $W_1(\Omega)$ (resp. $W_2(\mathsf{Plays}(\mathcal{A}) \setminus \Omega)$).

Given an n-player game $\mathcal{G} = (\mathcal{A}, (\mathsf{cost}_i)_{i \in [\![n]\!]})$ where $\mathcal{A} = ((V_i)_{i \in [\![n]\!]}, E)$, we define the *coalition game (for $\mathcal{P}_i$)* as the game opposing $\mathcal{P}_i$ to the coalition of the other players, formally defined as the two-player zero-sum game $\mathcal{G}_i = (\mathcal{A}_i, \mathsf{cost}_i)$ where $\mathcal{A}_i = ((V_i, V \setminus V_i), E)$. We write $\mathcal{P}_{-i}$ to refer to the coalition of players other than $\mathcal{P}_i$

We also refer to two-player zero-sum games where the objective (resp. cost function) of $\mathcal{P}_1$ is a reachability objective (resp. a truncated sum cost function, a Büchi objective) as reachability (resp. shortest path, Büchi) games.

3 Zero-sum games: punishing strategies

In this section, we present results on strategies in zero-sum games. They are of interest for the classical punishment mechanism used to construct NEs (described in Sect. 1). Intuitively, this mechanism functions as follows: if some player deviates from the intended outcome of the NE, the other players coordinate as a coalition to prevent the player from having a profitable deviation. The strategy of the coalition used to sabotage the deviating player is called a *punishing strategy*.

We explain that we can always find memoryless punishing strategies. First, we recall classical results on reachability and Büchi games. Second, we describe memoryless punishing strategies for shortest path games. We fix a two-player arena $\mathcal{A} = ((V_1, V_2), E)$ and a target $T \subseteq V$ for the remainder of this section.

Reachability and Büchi games. Zero-sum reachability games enjoy *memoryless determinacy*: they are determined and for both players, there exist *memoryless uniform winning strategies*. Furthermore, any vertex of $\mathcal{P}_2$ that is winning for $\mathcal{P}_2$ has a successor in this winning region. Any strategy of $\mathcal{P}_2$ that selects only such successors can be shown to be winning from any vertex in their winning region. The statements above follow, e.g., from the proof of [21, Proposition 2.18]. We summarise this information in the following theorem.

▶ **Theorem 4.** *Both players have memoryless uniform winning strategies in reachability games. Let $\mathcal{G} = (\mathcal{A}, \mathsf{Reach}(T))$, $W_2(\mathsf{Safe}(T))$ be the winning region of $\mathcal{P}_2$ in $\mathcal{G}$, $v_0 \in W_2(\mathsf{Safe}(T))$ and σ_2 be a strategy of $\mathcal{P}_2$. If for all histories $h \in \mathsf{Hist}_2(\mathcal{A})$ starting in v_0 containing only vertices of $W_2(\mathsf{Safe}(T))$, we have $\sigma_2(h) \in W_2(\mathsf{Safe}(T))$, then σ_2 is winning from v_0.*

Büchi games also enjoy memoryless determinacy. If follows from the memoryless determinacy of parity games [13], a class of objectives subsuming Büchi objectives.

▶ **Theorem 5.** *Both players have memoryless uniform winning strategies in Büchi games.*

Shortest path games. Let $w \colon E \to \mathbb{N}$ be a weight function and $\mathcal{G} = (\mathcal{A}, \mathsf{TS}_w^T)$ be a zero-sum shortest path game. First, we remark that $\mathcal{G}$ is determined. It can be shown using the determinacy of games with open objectives [16]. Furthermore, $\mathcal{P}_1$ has a memoryless uniform optimal strategy. Intuitively, this is because $\mathcal{P}_1$ has no need to remember the past, and only should follow a shortest path to a target from the current vertex. Although $\mathcal{P}_2$ does not necessarily have an optimal strategy (Ex. 3), it can be shown that there exists a family of $\mathcal{P}_2$ memoryless strategies labelled by non-negative integers that are winning from any vertex in the winning region of $\mathcal{P}_2$ in the reachability game $(\mathcal{A}, \mathsf{Reach}(T))$ and that ensure the minimum of the integer and the value of the vertex from any other vertex. This information is summarised in the following theorem. Its proof is provided in the full paper [20].

▶ **Theorem 6.** *The game $\mathcal{G}$ is determined. A memoryless uniform optimal strategy exists for $\mathcal{P}_1$. For all $\alpha \in \mathbb{N}$, there exists a memoryless strategy σ_2^α of $\mathcal{P}_2$ such that, for all $v \in V$: (i) σ_2^α is winning from v for $\mathcal{P}_2$ in the game $(\mathcal{A}, \mathsf{Reach}(T))$ if $v \in W_2(\mathsf{Safe}(T))$ and (ii) σ_2^α ensures a cost of at least* $\min\{\mathsf{val}(v), \alpha\}$.

4 Characterising Nash equilibria outcomes

We provide characterisations of plays that are outcomes of NEs in reachability, Büchi and shortest path games. These characterisations relate to the corresponding zero-sum games: they roughly state that a play is an NE outcome if and only if the cost incurred by a player from a vertex of the play is less than the value of said vertex in the coalition game opposing the player to the others. We provide a characterisation for reachability and Büchi games, and then a characterisation for shortest path games. We fix an arena $\mathcal{A} = ((V_i)_{i\in[\![n]\!]}, E)$ and targets $T_1, \dots, T_n \subseteq V$ for this entire section. Proofs of the results below are presented in the full paper [20].

Reachability and Büchi games. We first consider reachability and Büchi games: their respective NE outcome characterisations are close. Let $\mathcal{G} = (\mathcal{A}, (\Omega_i)_{i\in[\![n]\!]})$ be a reachability or Büchi game. We denote by $W_i(\Omega_i)$ the winning region of the first player of the coalition game $\mathcal{G}_i = (\mathcal{A}_i, \Omega_i)$, in which $\mathcal{P}_i$ is opposed to the other players. The characterisation follows. It relies on the existence of punishing strategies. An identical characterisation for finite arenas can be found in [11].

▶ **Theorem 7.** *Assume $\mathcal{G}$ is a reachability (resp. Büchi) game. Let $\pi = v_0v_1\dots$ be a play. Then π is the outcome of an NE from v_0 if and only if, for all $i \in [\![n]\!] \setminus \mathsf{VisPl}(\pi)$ (resp. $i \in [\![n]\!] \setminus \mathsf{InfPl}(\pi)$), $v_\ell \notin W_i(\mathsf{Reach}(T_i))$ (resp. $v_\ell \notin W_i(\mathsf{Büchi}(T_i))$) for all $\ell \in \mathbb{N}$.*

Shortest path games. Let $w\colon E \to \mathbb{N}$ be a weight function. We now consider a shortest path game $\mathcal{G} = (\mathcal{A}, (\mathsf{TS}_w^{T_i})_{i\in[\![n]\!]})$. For any $v \in V$, we denote by $\mathsf{val}_i(v)$ the value of v in the coalition game $\mathcal{G}_i = (\mathcal{A}_i, \mathsf{TS}_w^{T_i})$. We keep the notation $W_i(\mathsf{Reach}(T_i))$ of the previous section.

In reachability and Büchi games, Thm. 7 indicates that the value in coalition games (i.e., who wins) is sufficient to characterise NE outcomes. It is also the case in finite arenas for shortest path games [6, Theorem 15]. However, it is not in arbitrary arenas.

▶ **Example 8.** Let us consider the arena depicted in Fig. 1b and let $T_1 = \{t\}$ and $T_2 = \{v_0\}$. It holds that $\mathsf{val}_1(v_0) = +\infty$ (it follows from $\mathsf{val}_1(v_\infty) = +\infty$ which is shown in Ex. 3). Therefore, the cost of all suffixes of the play v_0^ω for $\mathcal{P}_1$ matches the value of their first vertex v_0. However, for any strategy profile resulting in v_0^ω from v_0, $\mathcal{P}_1$ has a profitable deviation in moving to v_∞ and using a reachability strategy to ensure a finite cost. ⌟

A value-based characterisation fails because of vertices $v \in W_i(\mathsf{Reach}(T_i))$ such that $\mathsf{val}_i(v)$ is infinite. Despite the infinite value of such vertices, $\mathcal{P}_i$ has a strategy such that their cost is finite no matter the behaviour of the others. To obtain a characterisation, we impose additional conditions on players whose targets are not visited that are related to reachability games. We obtain the following characterisation.

▶ **Theorem 9.** *Let $\pi = v_0v_1\dots$ be a play. Then π is an outcome of an NE from v_0 in $\mathcal{G}$ if and only (i) for all $i \in [\![n]\!] \setminus \mathsf{VisPl}(\pi)$ and $\ell \in \mathbb{N}$, we have $v_\ell \notin W_i(\mathsf{Reach}(T_i))$ and (ii) for all $i \in \mathsf{VisPl}(\pi)$ and all $\ell \leq r_i$, it holds that $\mathsf{TS}_w^{T_i}(\pi_{\geq \ell}) \leq \mathsf{val}_i(v_\ell)$ where $r_i = \min\{r \in \mathbb{N} \mid v_r \in T_i\}$.*

5 Finite-memory Nash equilibria in reachability games

In this section, we describe finite-memory strategy profiles for NEs with memory bounds that depend solely on the numbers of players in reachability and shortest path games. These finite-memory strategy profiles behave differently to those described for the characterisations in Thm. 7 and Thm. 9. Intuitively, following a deviation of $\mathcal{P}_i$, the coalition $\mathcal{P}_{-i}$ does not necessarily switch to a punishing strategy for $\mathcal{P}_i$. Instead, they may attempt to keep following a suffix of the equilibrium's original outcome if the deviation does not appear to prevent it.

This section is structured as follows. We illustrate the constructions for reachability and shortest path games with examples in Sect. 5.1. In Sect. 5.2, we provide templates for finite-memory NEs and technical notions to define them. In Sect. 5.3, we show that we can derive, from any NE outcome, another with a simple structure and provide the general constructions for finite-memory NEs with memory size independent of the arena. The details of the last two sections are provided in the full paper [20].

We fix an arena $\mathcal{A} = ((V_i)_{i \in [\![n]\!]}, E)$, target sets $T_1, \dots, T_n \subseteq V$ and a weight function $w \colon E \to \mathbb{N}$ for the remainder of this section. We introduce a new operator in this section. Given two histories $h = v_0 \dots v_\ell$ and $h' = v_\ell v_{\ell+1} \dots v_r$, we let $h \cdot h' = v_0 \dots v_\ell v_{\ell+1} \dots v_r$; we say that $h \cdot h'$ is the *combination* of h and h'. The combination $h \cdot \pi$ of a history h and a play π such that $\mathsf{last}(h) = \mathsf{first}(\pi)$ is defined similarly.

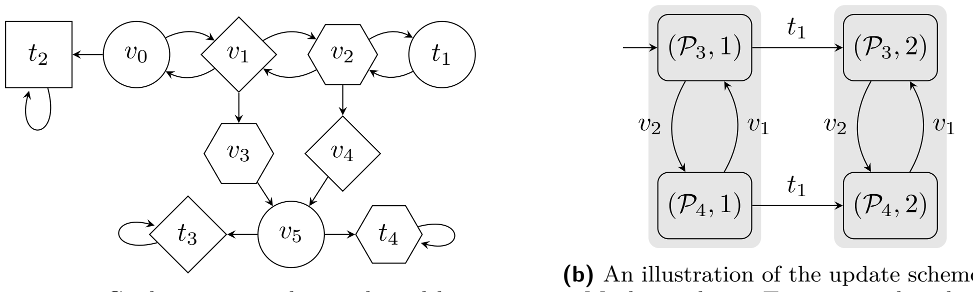
5.1 Examples

In this section, we illustrate the upcoming construction for finite-memory NEs for both settings of interest. We start with a reachability game.

▶ **Example 10.** We consider the game on the arena depicted in Fig. 2a where the objective of $\mathcal{P}_i$ is $\mathsf{Reach}(\{t_i\})$ for $i \in [\![4]\!]$. We present a finite-memory NE with outcome $\pi = v_0v_1v_2t_1v_2v_1v_0t_2^\omega$ to illustrate the idea behind the upcoming construction.

First, observe that π can be seen as the combination of the simple history $\mathsf{sg}_1 = v_0v_1v_2t_1$ and the simple lasso $\mathsf{sg}_2 = t_1v_2v_1v_0t_2^\omega$. The simple history sg_1 connects the initial vertex to the first visited target, and the simple lasso sg_2 connects the first target to the second and contains the suffix of the play. Therefore, if we were not concerned with the stability of the equilibrium, the outcome π could be obtained by using a finite-memory strategy profile where all strategies are defined by a Mealy machine with state space $[\![2]\!]$. Intuitively, these strategies would follow sg_1 while remaining in their first memory state 1, then, when t_1 is visited, they would update their memory state to 2 and follow sg_2.

We build on these simple Mealy machines with two states. We include additional information in each memory state. We depict a suitable Mealy machine state space and update scheme in Fig. 2b. The rectangles grouping together states $(\mathcal{P}_3, j)$ and $(\mathcal{P}_4, j)$



(a) An arena. Circles, squares, diamonds and hexagons are resp. $\mathcal{P}_1$, $\mathcal{P}_2$, $\mathcal{P}_3$, $\mathcal{P}_4$ vertices.

(b) An illustration of the update scheme of a Mealy machine. Transitions that do not change the memory state are omitted.

Figure 2 A reachability game and a representation of a Mealy machine update scheme suitable for an NE from v_0.

represent the memory state j of the simpler Mealy machine, for $j \in [\![2]\!]$. The additional information roughly encodes the last player to act among the players whose objective is not satisfied in π. More precisely, an update is performed from the memory state $(\mathcal{P}_i, j)$ only if the vertex fed to the Mealy machine appears in sg_j for $j \in [\![2]\!]$.

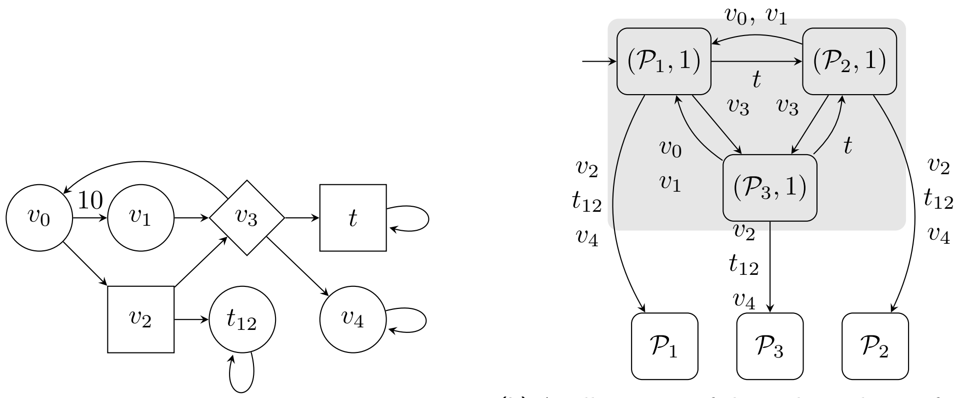
By construction, if $\mathcal{P}_i$ (among $\mathcal{P}_3$ and $\mathcal{P}_4$) deviates and exits the set of vertices of sg_j when in a memory state of the form $(\cdot, j)$, then the memory updates to $(\mathcal{P}_i, j)$ and does not change until the play returns to some vertex of sg_j (which is not possible here due to the structure of the graph, but may be in general). For instance, assume $\mathcal{P}_3$ moves from v_1 to v_3 after the history $h = v_0v_1v_2t_1v_2v_1$. Then the memory after h is in state $(\mathcal{P}_3, 2)$ and no longer changes from there on.

It remains to explain how the next-move function of the Mealy machine should be defined to ensure an NE. Essentially, for a state of the form $(\mathcal{P}_i, j)$ and vertices in sg_j, we assign actions as in the simpler two-state Mealy machine described previously. On the other hand, for a state of the form $(\mathcal{P}_i, j)$ and a vertex not in sg_j, we use a memoryless punishing strategy against $\mathcal{P}_i$. In this particular case, we need only specify what $\mathcal{P}_1$ should do in v_5. Naturally, in memory state $(\mathcal{P}_i, j)$, $\mathcal{P}_1$ should move to the target of the other player. It is essential to halt memory updates for vertices v_3 and v_4 to ensure the correct player is punished.

We close this example with comments on the structure of the Mealy machine. Assume the memory state is of the form $(\mathcal{P}_i, j)$. If a deviation occurs and leads to a vertex of sg_j other than the intended one, then the other players will continue trying to progress along sg_j and do not specifically try punishing the deviating player. Similarly, if after a deviation leaving the set of vertices of sg_j (from which point the memory is no longer updated until this set is rejoined), a vertex of sg_j is visited again, then the players resume trying to progress along this history and memory updates resume. In other words, these finite-memory strategies do not pay attention to all deviations and do not have dedicated memory that commit to punishing deviating players for the remainder of a play after a deviation. ⌟

We now give an example for the shortest path case. The Mealy machines we propose are slightly larger in this case. We argue that it may be necessary to commit to a punishing strategy if the set of vertices of the history the players want to progress along is left. This requires additional memory states. Our example shows that it may be necessary to punish deviations from players whose targets are visited, as they can possibly improve their cost.

▶ **Example 11.** We consider the shortest path game on the weighted arena depicted in Fig. 3a where the target of $\mathcal{P}_i$ is $T_i = \{t, t_{12}\}$ for $i \in [\![2]\!]$ and $T_3 = \{t\}$ for $\mathcal{P}_3$. We argue that a finite-memory NE with outcome $\pi = v_0v_1v_3t^\omega$ from v_0 cannot be obtained by adapting the construction of Ex. 10. We provide an alternative construction that builds on the same ideas.



(a) A weighted arena. Edge labels indicate their weight. Unlabelled edges have a weight of 1.

(b) An illustration of the update scheme of a Mealy machine. Transitions that do not change the memory state are omitted.

Figure 3 A shortest path game and a representation of a Mealy machine update scheme suitable for some NE from v_0.

In this case, π is a simple lasso, much like the second part of the play in the previous example. First, let us assume a Mealy machine similar to that of Ex. 10, i.e., such that it tries to progress along π whenever it is in one of its vertices. The update scheme of such a Mealy machine would be obtained by removing the transitions to states of the form $\mathcal{P}_i$ from Fig. 3b (replacing them by self-loops).

If $\mathcal{P}_3$ uses a strategy based on such a Mealy machine, then $\mathcal{P}_1$ has a profitable deviation from v_0: if $\mathcal{P}_1$ moves from v_0 to v_2, then either $\mathcal{P}_1$ incurs a cost of 2 if $\mathcal{P}_2$ moves to t_{12} from v_2 or a cost of 3 if $\mathcal{P}_2$ moves to v_3 as $\mathcal{P}_3$ would then move to t by definition of the Mealy machine. To circumvent this issue, if $\mathcal{P}_i$ exits the set of vertices of π, we update the memory to the punishment state $\mathcal{P}_i$. This results in the update scheme depicted in Fig. 3b. Next-move functions to obtain an NE can be defined as follows, in addition to the expected behaviour to obtain π: for $\mathcal{P}_2$, $\mathsf{nxt}_2((\mathcal{P}_1, 1), v_2) = \mathsf{nxt}_2(\mathcal{P}_1, v_2) = v_3$ and for $\mathcal{P}_3$, $\mathsf{nxt}_3(\mathcal{P}_1, v_3) = v_4$.

Similarly to the previous example, players do not explicitly react to deviations that move to vertices of π; if $\mathcal{P}_3$ deviates after reaching v_3 and moves back to v_0, the memory of the other players does not update to state $\mathcal{P}_3$. Intuitively, there is no need to switch to a punishing strategy for $\mathcal{P}_3$ as going back to the start of the intended outcome is more costly than conforming to it, preventing the existence of a profitable deviation.

This example differs slightly from the general construction below, which would decompose π into two parts: a history $v_0v_1v_3t$ from the initial vertex to the first target and the suffix t^ω of the play after all targets are visited. In full generality, this separation is needed [20]. ⌟

5.2 Segments and strategies

In Sect. 5.1, we illustrated that the finite-memory Nash equilibria we construct in reachability and shortest path games share a common structure. In this section, we provide the generic part of these Mealy machines. We first introduce decompositions of plays. We then partially define Mealy machines encoding strategies based on so-called simple decompositions.

Decomposing plays. We fix $\pi = v_0v_1 \ldots \in \mathsf{Plays}(\mathcal{A})$ for this whole section. We first introduce some terminology. A play or history is *simple* if no vertex occurs twice within. A play is a *simple lasso* if it is of the form pc^ω where $pc \in \mathsf{Hist}(\mathcal{A})$ is a simple history. A *segment* of π is either a suffix $\pi_{\geq \ell}$ of π ($\ell \in \mathbb{N}$) or any history of the form $v_\ell \dots v_{\ell'}$ ($\ell \leq \ell'$). We denote segments by sg to avoid distinguishing finite and infinite segments of plays in the following. A segment is *simple* if it is a simple history, a simple play or a simple lasso.

A (finite) *segment decomposition* of π is a sequence $\mathcal{D} = (\mathsf{sg}_1, \dots, \mathsf{sg}_k)$ where sg_j is a history for all $j < k$, sg_k is a suffix of π, $\mathsf{last}(\mathsf{sg}_j) = \mathsf{first}(\mathsf{sg}_{j+1})$ for all $j < k$ and $\pi = \mathsf{sg}_1 \cdot \dots \cdot \mathsf{sg}_k$. We assume that among the histories of a decomposition, there are none of the form $h = v$, i.e., there are no trivial segments. The segment decomposition $\mathcal{D}$ is *simple* if all segments within are simple. If there is some NE outcome with a given cost profile, we show that there is an NE outcome with a preferable cost profile that admits a simple segment decomposition. To obtain finite-memory NEs, we build on NE outcomes with a simple segment decomposition.

Finite-memory decomposition-based strategies. Let $\pi = v_0 v_1 \dots \in \mathsf{Plays}(\mathcal{A})$ be a play that admits a simple segment decomposition $\mathcal{D} = (\mathsf{sg}_1, \dots, \mathsf{sg}_k)$. We partially define a Mealy machine that serves as the basis for the finite-memory NEs described in the next section.

The memory state space is made of pairs of the form $(\mathcal{P}_i, j)$ for some $j \in [\![k]\!]$. We do not consider all such pairs, e.g., it is not necessary in Ex. 10. Therefore, we parameterise our construction by a non-empty set of players $I \subseteq [\![n]\!]$. We consider the memory state space $M^{I,\mathcal{D}} = \{\mathcal{P}_i \mid i \in I\} \times [\![k]\!]$. The initial state $m^{I,\mathcal{D}}_{\mathsf{init}}$ is any state of the form $(\mathcal{P}_i, 1) \in M^{I,\mathcal{D}}$.

The update function $\mathsf{up}^{I,\mathcal{D}}$ behaves similarly to Fig. 2b. It keeps track of the last player in I to have moved and the current segment. Formally, for any $(\mathcal{P}_i, j) \in M^{I,\mathcal{D}}$ and vertex v occurring in sg_j, we let $\mathsf{up}^{I,\mathcal{D}}((\mathcal{P}_i, j), v) = (\mathcal{P}_{i'}, j')$ where (i) i' is such that $v \in V_{i'}$ if $v \in \bigcup_{i'' \in I} V_{i''}$ and otherwise $i' = i$, and (ii) $j' = j + 1$ if $j < k$ and $v = \mathsf{last}(\mathsf{sg}_j)$ and $j' = j$ otherwise. Updates from $(\mathcal{P}_i, j)$ for a vertex that does not appear in sg_j are left undefined.

The next-move function $\mathsf{nxt}^{I,\mathcal{D}}_i$ of $\mathcal{P}_i$ proposes the next vertex of the current segment. Formally, given a memory state $(\mathcal{P}_{i'}, j) \in M^{I,\mathcal{D}}$ and a vertex $v \in V_i$ that occurs in sg_j, we let $\mathsf{nxt}^{I,\mathcal{D}}_i((\mathcal{P}_{i'}, j), v)$ be the vertex occurring after v in sg_{j+1} if $j < k$ and $v = \mathsf{last}(\mathsf{sg}_j)$, and otherwise we let it be the vertex occurring after v in sg_j. Like updates, the next-move function is left undefined in memory states $(\mathcal{P}_i, j)$ for a vertex that does not appear in sg_j.

5.3 Nash equilibria

We now present finite-memory NEs with memory bounds depending only on the number of players. We first derive, given an NE outcome, another NE outcome that admits a simple decomposition. We impose additional technical properties on these decompositions to define NEs with strategies based on them. We then define finite-memory strategies based on these simple decompositions by extending the partial definition above to obtain finite-memory NEs. We deal with reachability games then shortest paths games.

Simplifying outcomes. We explain that from any NE outcome in a shortest path game, we can derive another NE outcome with a preferable cost profile that admits a simple segment decomposition. The result extends to reachability games. We consider two cases.

First, we consider NE outcomes such that all players who see their target have the initial vertex of the outcome in it, generalising the case where no players see their targets. From these outcomes, we can directly derive an NE outcome that is a simple lasso or simple play.

▶ **Lemma 12.** *Let $\pi' \in \mathsf{Plays}(\mathcal{A})$ be the outcome of an NE from $v_0 \in V$ in a shortest path game $\mathcal{G} = (\mathcal{A}, (\mathsf{TS}^{T_i}_w)_{i \in [\![n]\!]})$ such that $\mathsf{VisPos}(\pi') \subseteq \{0\}$. There exists an NE outcome $\pi \in \mathsf{Plays}(\mathcal{A})$ from v_0 with the same cost profile as π' that is a simple lasso or a simple play and such that $\mathsf{VisPos}(\pi) \subseteq \{0\}$. In particular, π has the simple segment decomposition (π).*

We now consider NE outcomes such that some player sees their target later than in the initial vertex. In this case, we can derive an NE outcome with a simple decomposition such that the simple histories of the decomposition connect the first target elements that are visited. We impose a technical condition on these simple histories, to ensure that no player has a profitable deviation by skipping ahead in a segment.

▶ **Lemma 13.** *Let π' be the outcome of an NE from $v_0 \in V$ in a shortest path game $\mathcal{G} = (\mathcal{A}, (\mathsf{TS}_w^{T_i})_{i\in[\![n]\!]})$. Assume that $|\mathsf{VisPos}(\pi') \setminus \{0\}| = k > 0$. There exists an NE outcome π from v_0 with $\mathsf{VisPos}(\pi) \setminus \{0\} = \{\ell_1 < \ldots < \ell_k\}$ that admits a simple segment decomposition $(\mathsf{sg}_1, \ldots, \mathsf{sg}_{k+1})$ such that (i) $(\mathsf{sg}_1, \ldots, \mathsf{sg}_k \cdot \mathsf{sg}_{k+1})$ is also a simple decomposition of π; (ii) for all $j \in [\![k]\!]$, $\mathsf{sg}_1 \cdot \ldots \cdot \mathsf{sg}_j = \pi_{\leq \ell_j}$; (iii) for all $j \in [\![k]\!]$, $w(\mathsf{sg}_j)$ is minimum among all histories that share their first and last vertex with sg_j and traverse a subset of the vertices occurring in sg_j; and (iv) for all $i \in [\![n]\!]$, $\mathsf{TS}_w^{T_i}(\pi) \leq \mathsf{TS}_w^{T_i}(\pi')$.*

Reachability games. We fix a reachability game $\mathcal{G} = (\mathcal{A}, (\mathsf{Reach}(T_i))_{i\in[\![n]\!]})$. We construct finite-memory NEs by extending the partially-defined Mealy machines of Sect. 5.2 and generalising the strategies presented in Ex. 10. The main result is the following.

▶ **Theorem 14.** *Let σ' be an NE from a vertex v_0. There exists a finite-memory NE σ from v_0 such that $\mathsf{VisPl}(\mathsf{Out}(\sigma, v_0)) = \mathsf{VisPl}(\mathsf{Out}(\sigma', v_0))$ where each strategy of σ has a memory size of at most n^2. Precisely, a memory size of $\max\{1, n - |\mathsf{VisPl}(\mathsf{Out}(\sigma', v_0))|\} \cdot \max\{1, |\mathsf{VisPos}(\mathsf{Out}(\sigma', v_0)) \setminus \{0\}|\}$ suffices.*

Proof sketch. Consider an NE outcome π and its simple decomposition $\mathcal{D} = (\mathsf{sg}_1, \ldots, \mathsf{sg}_k)$ provided by Lem. 12 and Lem. 13 (point (i)) for $\mathsf{Out}(\sigma', v_0)$. The general idea of the construction is to use the state space $M^{I,\mathcal{D}}$ where $I \subseteq [\![n]\!]$ is the set of players who do not see their targets if it is non-empty, or a single arbitrary player if all players see their target. Let $i' \in I$ and $j \in [\![k]\!]$. We extend $\mathsf{up}^{I,\mathcal{D}}$ to leave unchanged the memory state if in state $(\mathcal{P}_{i'}, j)$ whenever the current vertex is not in sg_j. In this same situation, the next-move functions $\mathsf{nxt}_i^{I,\mathcal{D}}$ are extended to assign moves from a uniform memoryless winning strategy of the second player in the coalition game $\mathcal{G}_{i'} = (\mathcal{A}_{i'}, \mathsf{Reach}(T_{i'}))$ (which exists by Thm. 4). The equilibrium's stability is a consequence of Thm. 7 and the second statement of Thm. 4. ◀

We remark that Thm. 14 provides a memory bound that is *linear in the number of players* when no players see their target and when all players see their target.

▶ **Corollary 15.** *If there exists an NE from v_0 such that no (resp. all) players see their target in its outcome, then there is a finite-memory NE from v_0 such that no (resp. all) players see their target in its outcome such that all strategies have a memory size of at most n.*

Shortest path games. We now fix a shortest path game $\mathcal{G} = (\mathcal{A}, (\mathsf{TS}_w^{T_i})_{i\in[\![n]\!]})$. We provide an alternative generalisation of the partially-defined Mealy machines described in Sect. 5.2, this time generalising the strategies provided in Ex. 11. Ex. 11 shows that only altering the construction of Thm. 14 to also monitor (and punish) players whose targets are visited is not sufficient. To overcome this, we change the approach so players commit to punishing any player who exits the current segment of the intended outcome.

▶ **Theorem 16.** *Let σ' be an NE from a vertex v_0. There exists a finite-memory NE σ from v_0 such that $\mathsf{VisPl}(\mathsf{Out}(\sigma, v_0)) = \mathsf{VisPl}(\mathsf{Out}(\sigma', v_0))$ and, for all $i \in [\![n]\!]$, $\mathsf{TS}_w^{T_i}(\mathsf{Out}(\sigma, v_0)) \leq \mathsf{TS}_w^{T_i}(\mathsf{Out}(\sigma', v_0))$ where each strategy of σ has a memory size of at most $n^2 + 2n$. Precisely, a memory size of $n \cdot (|\mathsf{VisPos}(\mathsf{Out}(\sigma', v_0)) \setminus \{0\}| + 2)$ suffices.*

Proof sketch. The construction is similar to that of Thm. 14. We obtain a suitable NE outcome π and its decomposition $\mathcal{D}$ in the same way, and build on $M^{I,\mathcal{D}}$, $\mathsf{up}^{I,\mathcal{D}}$, $\mathsf{nxt}_i^{I,\mathcal{D}}$ with $I = [\![n]\!]$. If we apply Lem. 13 here, we consider the first decomposition the lemma provides and not the one from point (i) of the lemma. Intuitively, merging the last two segments, as done in the decomposition from Lem. 13, point (i), prevents players from reacting to deviations within the merged segment and could enable a profitable deviation.

In this case, instead of freezing memory updates if the current segment is left when the memory state is of the form $(\mathcal{P}_i, j)$, the memory switches to a memory state $\mathcal{P}_i$ that is never left. This switch can only occur if $\mathcal{P}_i$ deviates. The next-move function, for this memory state, assigns moves from a punishing strategy obtained from the coalition game $\mathcal{G}_i = (\mathcal{A}_i, \mathsf{TS}_w^{T_i})$ by Thm. 6, chosen to hinder $\mathcal{P}_i$, ensuring that in case of a deviation, $\mathcal{P}_i$'s cost is at least that of the original outcome.

The conditions imposed on outcomes of Lem. 13 (notably condition (iii)) and the characterisation of Thm. 9 imply the correctness of this construction. Condition (iii) of Lem. 13 ensures that a player cannot reach their target with a lesser cost by traversing the vertices within a segment in another order, whereas the characterisation of Thm. 9 guarantees that the punishing strategies sabotage deviating players sufficiently. ◀

In this case, Thm. 16 provides the memory bound $2n$ if no players visit their target. However, the construction of Thm. 9 applies to such NEs in shortest path games.

▶ **Corollary 17.** *If there exists an NE from v_0 such that no players see their target in its outcome, then there is a finite-memory NE from v_0 such that no players see their target in its outcome such that all strategies have a memory size of at most n.*

6 Finite-memory Nash equilibria in Büchi games

We now present finite-memory NEs for Büchi games. We illustrate in Sect. 6.1 that the constructions for reachability and shortest path games do not extend directly to Büchi games. We build on the techniques of Sect. 5.2 to provide finite-memory NEs in Sect. 6.2. We fix an arena $\mathcal{A} = ((V_i)_{i\in[\![n]\!]}, E)$, targets $T_1, \ldots, T_n \subseteq V$ and the Büchi game $\mathcal{G} = (\mathcal{A}, (\mathsf{Büchi}(T_i))_{i\in[\![n]\!]})$ for this entire section. Proofs and details of this section are presented in the full version of the paper [20].

6.1 Examples

For reachability and shortest path games, we relied on simple segment decompositions between consecutive targets along some NE outcome to obtain finite-memory NEs. Our strategies based on these decompositions do not explicitly punish players who deviate. We show that this can be problematic when dealing with Büchi objectives.

▶ **Example 18.** Consider the game on the arena depicted in Fig. 4a where the objectives of $\mathcal{P}_1$ and $\mathcal{P}_2$ are $\mathsf{Büchi}(\{v_1\})$ and $\mathsf{Büchi}(\{v_2\})$ respectively. The play $v_0v_1v_2^\omega$ is the outcome of an NE by Thm. 7. To mimic the construction underlying Thm. 14 and Thm. 16, we would consider a finite-memory strategy based on the decomposition $\mathcal{D} = (v_0v_1v_2, v_2^\omega)$. However, if $\mathcal{P}_2$ uses such a strategy, $\mathcal{P}_1$ would enforce their objective via the memoryless strategy σ_1 such that $\sigma_2(v_1) = v_0$, resulting in the outcome $(v_0v_1)^\omega$, as $\mathcal{P}_2$ would not punish the deviation. ⌟

In the previous example, the issue with the proposed decomposition lies with the occurrence of a target of $\mathcal{P}_2$, whose objective is not satisfied in the intended outcome, within some segment of the decomposition. To circumvent this issue, we construct strategies that

(a) An arena where a direct segment-based approach fails to obtain an NE.

(b) An arena on which players should commit to punishing strategies once a segment is left.

Figure 4 Two arenas. Circle, squares and diamonds are resp. $\mathcal{P}_1$, $\mathcal{P}_2$ and $\mathcal{P}_3$ vertices.

follow two phases in the following section. In their first phase, these strategies punish any deviations from the intended outcome. For their second phase, we adapt the strategies of Section 5.2. To ensure no profitable deviations may appear in the second phase, we start it at a point of the intended outcome from which no more targets of losing players occur.

We close this section by illustrating that the punishing mechanism used for finite-memory NEs in reachability games does not suffice, i.e., players must commit to their punishing strategies once some player exits the current segment in the second phase mentioned above.

▶ **Example 19.** Consider the game on the arena depicted in Fig. 4b where the objectives of $\mathcal{P}_1$, $\mathcal{P}_2$ and $\mathcal{P}_3$ are $\mathsf{Büchi}(\{v_1\})$ and $\mathsf{Büchi}(\{v_2, v_4\})$ and $\mathsf{Büchi}(\{v_4\})$ respectively. The play $\pi = (v_0v_1v_3)^\omega$ is the outcome of an NE by Thm. 7. Consider a $\mathcal{P}_1$ strategy based on the decomposition (π) that uses the punishment mechanism we introduced for reachability games. Then the behaviour of $\mathcal{P}_1$ does not change if $\mathcal{P}_2$ moves from v_0 to v_2 instead of v_1: $\mathcal{P}_1$ would move from v_1 to v_3 and then to v_0. It follows that $\mathcal{P}_2$ would have a profitable deviation no matter the strategy of $\mathcal{P}_3$.

To obtain an NE where all players use strategies based on the decomposition (π), $\mathcal{P}_1$, must commit to a punishing strategy for $\mathcal{P}_2$ if v_2 is visited. For $\mathcal{P}_2$ and $\mathcal{P}_3$ we consider the memoryless strategies σ_2 and σ_3 such that $\sigma_2(v_0) = \sigma_3(v_2) = v_1$. It is easy to check that this is an NE. ⌟

6.2 Finite-memory Nash equilibria

In this section, we establish the counterpart of Thm. 14 and Thm. 16 for Büchi games. It is split in two statements (Thm. 21 and Thm. 23) that depend on the form of the outcome of the considered NE. Each case is considered in a dedicated section. First, we consider NE outcomes with a vertex that occurs infinitely often within. We then show the result for NE outcomes without infinitely occurring vertices. For both cases, we first provide NE outcomes with a simple structure and then construct corresponding finite-memory NEs.

We consider alternative segment decompositions in this section. These decompositions differ from those defined in Sect. 5 in the following way. First, we allow *infinite* segment decompositions and tolerate decompositions such that their first segment is trivial. We extend the definition of simple segment to include simple cycles.

Throughout this section, we assume without loss of generality that any considered NE outcome π is such that $\mathsf{InfPl}(\pi)$ is not empty. This can be ensured by adding a new player for whom all vertices are targets if necessary.

Outcomes with an infinitely occurring vertex. The first case we consider is a generalisation of the finite-arena case: in a finite arena, all plays contain some infinitely occurring vertex. To obtain finite-memory NEs, we use the two-phase mechanism presented previously with an adaptation of the decomposition-based finite-memory strategies of Section 5.2 that can handle infinite ultimately periodic decompositions.

The following lemma provides NE outcomes with a simple structure on which we rely to define finite-memory NEs.

▶ **Lemma 20.** *Let π' be the outcome of an NE from $v_0 \in V$ in the Büchi game $\mathcal{G}$ such that some vertex occurs infinitely often in π' and let $k = |\mathsf{InfPl}(\pi')|$. Then there exists an NE outcome π from v_0 with $\mathsf{InfPl}(\pi) = \mathsf{InfPl}(\pi')$ such that π admits an infinite simple segment decomposition $(\mathsf{sg}_0, \mathsf{sg}_1, \ldots)$ such that (i) for all $j \geq 1$ and all $i \in [\![n]\!] \setminus \mathsf{InfPl}(\pi)$, no vertex of T_i occurs in sg_j and (ii) for all $j \geq 1$, $\mathsf{sg}_j = \mathsf{sg}_{j+k}$.*

We now state the main theorem of this section.

▶ **Theorem 21.** *Let σ' be an NE from a vertex v_0 such that some vertex occurs infinitely often in its outcome. There exists a finite-memory NE σ from v_0 such that $\mathsf{InfPl}(\mathsf{Out}(\sigma, v_0)) = \mathsf{InfPl}(\mathsf{Out}(\sigma', v_0))$. If $\mathcal{A}$ is finite, a memory size of at most $|V| + n^2 + n$ suffices.*

Proof sketch. Let π and $\mathcal{D} = (\mathsf{sg}_0, \mathsf{sg}_1, \ldots)$ be given by Lem. 20 for $\mathsf{Out}(\sigma', v_0)$ and let $k = |\mathsf{InfPl}(\pi)|$. We obtain finite-memory NEs via the the two-phase finite-memory strategies described in Sect. 6.1. For the first phase, we follow the history sg_0. For the second phase, we switch to a strategy that is based on the decomposition $\mathcal{D}' = (\mathsf{sg}_1, \mathsf{sg}_2, \ldots)$. Although this decomposition is infinite, we can construct a finite-memory strategy based on $\mathcal{D}$ by exploiting its ultimately periodic nature. To achieve this, we alter the definitions of Sect. 5.2: when reading $\mathsf{last}(\mathsf{sg}_k)$ in memory states of the form $(\mathcal{P}_i, k)$, we update the memory to an appropriate memory state of the form $(\mathcal{P}_{i'}, 1)$.

By completing the behaviour described above with switches to memoryless punishing strategies (Thm. 5) if sg_0 is not accurately simulated or if a player exits the current segment, we obtain a finite-memory NE. The stability of the NE follows from Thm. 7 for deviations that induce the use of punishing strategies and the property that no targets of losing players occur in segments sg_j, $j \geq 1$ for other deviations. ◀

With the classical approach to derive NEs from outcomes with a finite representation (Sect. 1), we can also design finite-memory NEs for outcomes obtained by Lem. 20. If $|V|$ is finite, the resulting strategies of this approach have a memory size of at most $(|V| + 2)n$. It follows our construction is preferable if there are few players compared to vertices.

Outcomes without an infinitely occurring vertex. We now deal with NE outcomes that can only appear in infinite arenas. We once again rely on a two-phase mechanism where the first phase is unchanged. The second phase is loosely based on an infinite decomposition. Intuitively, we allocate infinitely many disjoint segments to a same group of memory state. Due to this, players may not react to someone exiting the current segment.

The following lemma is the counterpart of Lemma 20 for this case.

▶ **Lemma 22.** *Let π' be the outcome of an NE from $v_0 \in V$ in the Büchi game $\mathcal{G}$ such that no vertex occurs infinitely often in π'. Then there exists an NE outcome π from v_0 with $\mathsf{InfPl}(\pi) = \mathsf{InfPl}(\pi')$ such that π admits an infinite simple segment decomposition $(\mathsf{sg}_0, \mathsf{sg}_1, \ldots)$ such that (i) for all $j \geq 1$ and all $i \in [\![n]\!] \setminus \mathsf{InfPl}(\pi)$, no vertex of T_i occurs in sg_j and (ii) for all $j \neq j'$, sg_j and $\mathsf{sg}_{j'}$ have no vertices in common if j and j' have the same parity.*

We can now state the last theorem of this section.

▶ **Theorem 23.** *Let σ' be an NE from a vertex v_0 such that no vertex occurs infinitely often in its outcome. There exists a finite-memory NE σ from v_0 such that $\mathsf{InfPl}(\mathsf{Out}(\sigma, v_0)) = \mathsf{InfPl}(\mathsf{Out}(\sigma', v_0))$.*

Proof sketch. Let π be the play and $\mathcal{D} = (\mathsf{sg}_0, \mathsf{sg}_1, \ldots)$ be the decomposition of π given by Lem. 22 for $\mathsf{Out}(\sigma', v_0)$. We rely once again on strategies with two phases. The first phase is defined exactly as for Thm. 21. For the second phase, we also adapt the definitions of Sect. 5.2. The update and next-move function in the original definitions are defined for each memory state of the form $(\mathcal{P}_i, j)$ based on the segment sg_j. In this case, we define the update and next-move functions in memory states of the form $(\mathcal{P}_i, 1)$ (resp. $(\mathcal{P}_i, 2)$) based on all odd segments (resp. all even segments besides sg_0) of $\mathcal{D}$ simultaneously, such that when the end of an even segment is reached in a memory state of the form $(\mathcal{P}_i, 2)$, the memory is updated to a state of the form $(\mathcal{P}_{i'}, 1)$. The fact all odd (resp. even) segments traverse pairwise disjoint set of vertices ensures that the next-move function is well-defined.

If at some point in the second phase, a vertex that does not occur in an odd segment is read in a memory state $(\mathcal{P}_i, 1)$, the memory is updated to a punishing state $\mathcal{P}_i$, such that players attempt to punish $\mathcal{P}_i$ with a memoryless strategy (Thm. 5). We proceed similarly for the even case. The resulting finite-memory strategy profile is an NE from v_0. On the one hand, any deviation such that the memory never updates to a punishing state must only have vertices that occur in segment sg_j with $j \neq 0$ in the limit. By choice of $\mathcal{D}$, this deviation cannot be profitable. Otherwise, it can be argued that the punishing strategy does in fact sabotage the deviating player, so long as their objective is not satisfied in π, by Thm. 7. ◀

References

1 Roderick Bloem, Krishnendu Chatterjee, and Barbara Jobstmann. Graph games and reactive synthesis. In Edmund M. Clarke, Thomas A. Henzinger, Helmut Veith, and Roderick Bloem, editors, *Handbook of Model Checking*, pages 921–962. Springer, 2018. doi:10.1007/978-3-319-10575-8_27.

2 Patricia Bouyer, Romain Brenguier, Nicolas Markey, and Michael Ummels. Pure Nash equilibria in concurrent deterministic games. *Log. Methods Comput. Sci.*, 11(2), 2015. doi:10.2168/LMCS-11(2:9)2015.

3 Patricia Bouyer, Stéphane Le Roux, Youssouf Oualhadj, Mickael Randour, and Pierre Vandenhove. Games where you can play optimally with arena-independent finite memory. *Log. Methods Comput. Sci.*, 18(1), 2022. doi:10.46298/lmcs-18(1:11)2022.

4 Patricia Bouyer, Mickael Randour, and Pierre Vandenhove. Characterizing omega-regularity through finite-memory determinacy of games on infinite graphs. *TheoretiCS*, 2, 2023. doi:10.46298/theoretics.23.1.

5 Thomas Brihaye, Véronique Bruyère, Aline Goeminne, Jean-François Raskin, and Marie van den Bogaard. The complexity of subgame perfect equilibria in quantitative reachability games. *Log. Methods Comput. Sci.*, 16(4), 2020. URL: https://lmcs.episciences.org/6883.

6 Thomas Brihaye, Véronique Bruyère, Aline Goeminne, and Nathan Thomasset. On relevant equilibria in reachability games. *J. Comput. Syst. Sci.*, 119:211–230, 2021. doi:10.1016/j.jcss.2021.02.009.

7 Thomas Brihaye, Julie De Pril, and Sven Schewe. Multiplayer cost games with simple Nash equilibria. In Sergei N. Artëmov and Anil Nerode, editors, *Logical Foundations of Computer Science, International Symposium, LFCS 2013, San Diego, CA, USA, January 6-8, 2013. Proceedings*, volume 7734 of *Lecture Notes in Computer Science*, pages 59–73. Springer, 2013. doi:10.1007/978-3-642-35722-0_5.

8 Thomas Brihaye, Aline Goeminne, James C. A. Main, and Mickael Randour. Reachability games and friends: A journey through the lens of memory and complexity (invited talk). In Patricia Bouyer and Srikanth Srinivasan, editors, *43rd IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2023, December 18-20, 2023, IIIT Hyderabad, Telangana, India*, volume 284 of *LIPIcs*, pages 1:1–1:26. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICS.FSTTCS.2023.1.

9 Véronique Bruyère. Computer aided synthesis: A game-theoretic approach. In Émilie Charlier, Julien Leroy, and Michel Rigo, editors, *Developments in Language Theory - 21st International Conference, DLT 2017, Liège, Belgium, August 7-11, 2017, Proceedings*, volume 10396 of *Lecture Notes in Computer Science*, pages 3–35. Springer, 2017. `doi:10.1007/978-3-319-62809-7_1`.

10 Véronique Bruyère, Noémie Meunier, and Jean-François Raskin. Secure equilibria in weighted games. In Thomas A. Henzinger and Dale Miller, editors, *Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), CSL-LICS '14, Vienna, Austria, July 14 - 18, 2014*, pages 26:1–26:26. ACM, 2014. `doi:10.1145/2603088.2603109`.

11 Rodica Condurache, Emmanuel Filiot, Raffaella Gentilini, and Jean-François Raskin. The complexity of rational synthesis. In Ioannis Chatzigiannakis, Michael Mitzenmacher, Yuval Rabani, and Davide Sangiorgi, editors, *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, volume 55 of *LIPIcs*, pages 121:1–121:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016. `doi:10.4230/LIPICS.ICALP.2016.121`.

12 Julie De Pril. *Equilibria in Multiplayer Cost Games*. PhD thesis, UMONS, 2013. URL: `http://math.umons.ac.be/staff/ancien/DePril.Julie/thesis_Julie_DePril.pdf`.

13 E. Allen Emerson and Charanjit S. Jutla. The complexity of tree automata and logics of programs. In *FOCS*, pages 328–337. IEEE Computer Society, 1988. `doi:10.1109/SFCS.1988.21949`.

14 Nathanaël Fijalkow, Nathalie Bertrand, Patricia Bouyer-Decitre, Romain Brenguier, Arnaud Carayol, John Fearnley, Hugo Gimbert, Florian Horn, Rasmus Ibsen-Jensen, Nicolas Markey, Benjamin Monmege, Petr Novotný, Mickael Randour, Ocan Sankur, Sylvain Schmitz, Olivier Serre, and Mateusz Skomra. Games on graphs. *CoRR*, abs/2305.10546, 2023. `doi:10.48550/arXiv.2305.10546`.

15 James Friedman. A non-cooperative equilibrium for supergames. *Review of Economic Studies*, 38(1):1–12, 1971. URL: `https://EconPapers.repec.org/RePEc:oup:restud:v:38:y:1971:i:1:p:1-12`.

16 David Gale and Frank M Stewart. Infinite games with perfect information. *Contributions to the Theory of Games*, 2(245-266):2–16, 1953.

17 Hugo Gimbert and Wieslaw Zielonka. Games where you can play optimally without any memory. In *CONCUR 2005 - Concurrency Theory, 16th International Conference, CONCUR 2005, San Francisco, CA, USA, August 23-26, 2005, Proceedings*, pages 428–442, 2005. `doi:10.1007/11539452_33`.

18 Erich Grädel, Wolfgang Thomas, and Thomas Wilke, editors. *Automata, Logics, and Infinite Games: A Guide to Current Research [outcome of a Dagstuhl seminar, February 2001]*, volume 2500 of *Lecture Notes in Computer Science*. Springer, 2002. `doi:10.1007/3-540-36387-4`.

19 Stéphane Le Roux and Arno Pauly. Extending finite-memory determinacy to multi-player games. *Inf. Comput.*, 261:676–694, 2018. `doi:10.1016/j.ic.2018.02.024`.

20 James C. A. Main. Arena-independent memory bounds for Nash equilibria in reachability games. *CoRR*, abs/2310.02142, 2023. `doi:10.48550/ARXIV.2310.02142`.

21 René Mazala. Infinite games. In Erich Grädel, Wolfgang Thomas, and Thomas Wilke, editors, *Automata, Logics, and Infinite Games: A Guide to Current Research [outcome of a Dagstuhl seminar, February 2001]*, volume 2500 of *Lecture Notes in Computer Science*, pages 23–42. Springer, 2001. `doi:10.1007/3-540-36387-4_2`.

22 John F. Nash. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences*, 36(1):48–49, 1950. `doi:10.1073/pnas.36.1.48`.

23 Martin J. Osborne and Ariel Rubinstein. *A course in game theory*. The MIT Press, 1994.

24 Mickael Randour. Automated synthesis of reliable and efficient systems through game theory: A case study. In *Proc. of ECCS 2012*, Springer Proceedings in Complexity XVII, pages 731–738. Springer, 2013. `doi:10.1007/978-3-319-00395-5_90`.

25 Michael Ummels. Rational behaviour and strategy construction in infinite multiplayer games. In S. Arun-Kumar and Naveen Garg, editors, *FSTTCS 2006: Foundations of Software Technology and Theoretical Computer Science, 26th International Conference, Kolkata, India, December 13-15, 2006, Proceedings*, volume 4337 of *Lecture Notes in Computer Science*, pages 212–223. Springer, 2006. doi:10.1007/11944836_21.

26 Michael Ummels. The complexity of Nash equilibria in infinite multiplayer games. In Roberto M. Amadio, editor, *Foundations of Software Science and Computational Structures, 11th International Conference, FOSSACS 2008, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2008, Budapest, Hungary, March 29 - April 6, 2008. Proceedings*, volume 4962 of *Lecture Notes in Computer Science*, pages 20–34. Springer, 2008. doi:10.1007/978-3-540-78499-9_3.

27 Michael Ummels and Dominik Wojtczak. The complexity of Nash equilibria in limit-average games. In Joost-Pieter Katoen and Barbara König, editors, *CONCUR 2011 - Concurrency Theory - 22nd International Conference, CONCUR 2011, Aachen, Germany, September 6-9, 2011. Proceedings*, volume 6901 of *Lecture Notes in Computer Science*, pages 482–496. Springer, 2011. doi:10.1007/978-3-642-23217-6_32.