Games with ω-Automatic Preference Relations

Véronique Bruyère
Université de Mons (UMONS), Belgium

Christophe Grandmont
Université de Mons (UMONS), Belgium
Université libre de Bruxelles (ULB), Belgium

Jean-François Raskin
Université libre de Bruxelles (ULB), Belgium

Abstract

This paper investigates Nash equilibria (NEs) in multi-player turn-based games on graphs, where player preferences are modeled as ω-automatic relations via deterministic parity automata. Unlike much of the existing literature, which focuses on specific reward functions, our results apply to any preference relation definable by an ω-automatic relation. We analyze the computational complexity of determining the existence of an NE (possibly under some constraints), verifying whether a given strategy profile forms an NE, and checking whether a specific outcome can be realized by an NE. When a (constrained) NE exists, we show that there always exists one with finite-memory strategies. Finally, we explore fundamental properties of ω-automatic relations and their implications in the existence of equilibria.

2012 ACM Subject Classification Theory of computation → Automata over infinite objects; Software and its engineering → Formal methods; Theory of computation → Solution concepts in game theory; Theory of computation → Exact and approximate computation of equilibria

Keywords and phrases Games played on graphs, Nash equilibrium, ω-automatic relations, ω-recognizable relations, constrained Nash equilibria existence problem

Digital Object Identifier 10.4230/LIPIcs.MFCS.2025.31

Related Version *Full Version*: https://arxiv.org/abs/2503.04759 [15]

Funding This work has been supported by the Fonds de la Recherche Scientifique – FNRS under Grant n° T.0023.22 (PDR Rational).
Jean-François Raskin: Supported by Fondation ULB (https://www.fondationulb.be/en/) and the Thelam Fondation.

1 Introduction

Non-zero-sum games on graphs provide a powerful framework for analyzing rational behavior in multi-agent systems, see, e.g., [9, 10, 14, 22, 27, 30, 33]. By modeling settings where agents have individual objectives, this approach captures the complexity of real-world scenarios where the interests of agents (modeled by players) are neither fully aligned nor entirely antagonistic. It enables the study of solution concepts such as *Nash* and *subgame-perfect equilibria* [36, 37], offering insight into strategic decision making. This, in turn, can aid in designing systems that anticipate and respond to rational behaviors, enriching reactive synthesis methodologies.

In this context, specifying player *objectives* [26] is central to reasoning about strategies and equilibria. In qualitative games, objectives determine whether an execution (an infinite path in the graph) is winning or losing for a given player. In quantitative games, executions are instead assigned numerical values, allowing players to compare and rank them based on accumulated rewards, with higher values being preferable. From this perspective, the qualitative setting can be viewed as a special case where the values are Boolean, typically captured by parity acceptance conditions, which encompass all ω-regular objectives. In the quantitative setting, a variety of reward functions have been explored, including total sum, limsup and liminf, discounted-sum, and hybrid models such as cost-optimal reachability. For each of these functions, dedicated techniques have been developed to design algorithms that analyze optimal strategies and, more broadly, equilibria.

However, these solutions are often tightly coupled to the specific reward function used, which limits their generality. When a new reward function or combination thereof is introduced, significant technical effort is required, as existing techniques rarely transfer across different reward models. This lack of general results – where solutions remain specialized to the underlying evaluation model, preventing knowledge transfer between different classes of objectives – has been noted in related contexts such as quantitative verification (see, e.g., [2]).

To address this, we propose a general approach based on *automata-based* preference relations to compare infinite paths in the graph. This framework provides a structured and unified method for reasoning about strategies and equilibria across various reward models. A similar use of automata-based preference relations has been explored in [2, 4, 8], and here we demonstrate how this idea can be adapted to fit the non-zero-sum game setting.

Contributions. Our contributions center on using *ω-automatic relations* on infinite words [41], as introduced in [23], to define a general framework for preference relations over paths in game graphs, thereby establishing a generic method to compare executions for players in non-zero-sum games. These relations are specified by deterministic parity automata that read pairs (x, y) of words synchronously and accept them whenever y is preferred to x.

Our main contributions focus on the computational complexity of *four key problems* related to NEs in non-zero-sum games [36] with ω-automatic preference relations. First, we study the problem of *verifying* whether a given strategy profile, specified by Mealy machines, one per strategy, constitutes an NE in the given game. We prove that this problem is PSPACE-complete (Theorem 3). Second, we examine whether a lasso-shaped path (i.e., a regular path) is the *outcome of an NE*, showing that this problem is in NP ∩ coNP and Parity-hard (Theorem 4). Third, we establish the existence of games without any NE, motivating the fundamental problem of determining whether a given game *admits at least one NE*. This problem turned out to be particularly challenging, and we reduce it to a three-player zero-sum game with imperfect information. We provide an algorithm for solving this problem with exponential complexity in the size of the graph, the parity automata defining the preference relation, and the number of their priorities, and doubly exponential complexity in the number of players. However, since the number of players is a natural parameter that tends to be small in practical scenarios,[1] we refine this result by proving that for a fixed number of players, the problem lies in EXPTIME and is PSPACE-hard (Theorem 5). In addition, our approach has the advantage of being modular and therefore easily adapts to question the existence of a *constrained NE*. When we attach one constraint to each player given as a lasso-shaped path and ask for an NE whose outcome is preferred to any of those constraints, the adapted algorithm keeps the same complexity except that it becomes doubly exponential in the number of priorities of the parity conditions. Yet the number of priorities is often

[1] In robotic systems or in security protocols, the number of agents is usually limited to a few. For example, in a security protocol, the players are Alice and Bob who exchange messages, the trusted third party, and a fourth player for the network (see, e.g., [19, 29])

small[2] and when we fix it and the number of players, the algorithm remains in EXPTIME and PSPACE-hard (Theorem 6). Note that our approach allows to show that when there exists an (constrained) NE, there exists one composed of finite-memory strategies.

Additionally, we analyze the algorithmic complexity of verifying whether an ω-automatic relation satisfies the axioms of a *strict partial order* (irreflexivity and transitivity) or of a *preorder* (reflexivity and transitivity) which are two classical requirements for a relation to model preferences. We show that these problems are NL-complete (Proposition 9). Finally, we show that when the ω-automatic preference relations are all *ω-recognizable* (a strict subclass of ω-automatic relations where the two input words can be processed independently) and preorders, the existence of at least one NE is always guaranteed (Theorem 11).

Related work. A well-established *hierarchy of rational relations* holds for both finite and infinite words [17, 40]. The ω-automatic relations – also called synchronized ω-rational relations – were first studied in [23]. Some decision problems about ω-automatic and ω-recognizable relations were solved in [35] and improved in [3]. The study of *automatic structures* has also led to results involving rational relations, notably within first-order logic (see, e.g., [5, 25, 31, 39]).

The problems we study in this paper were widely investigated in the literature for *specific* reward functions, including functions that mix different objectives, see, e.g., [6, 28, 42, 43]. There are also works that study these problems across *large* classes of reward functions rather than individual ones, or that consider *general* notions of preference relations. For instance, in [12], the authors prove the existence of finite-memory NEs for all cost-semi-linear reward functions. In [7], a complete methodology is developed to solve the (constrained) NE existence problem, thanks to the concept of suspect game, encompassing all reward functions definable by a class of monotone circuits over the set of states that appear (finitely or infinitely often) along paths in a game graph. The preference relations studied in [7] are all ω-automatic. In [20], the authors study NEs for games with a reward function that, given a finite set X of objectives of the same type, associates an integer with each subset of satisfied objectives of X. Again, if the objectives of X are ω-regular, the reward functions of [20] lead to ω-automatic preference relations. The existence of NEs is guaranteed within a broad setting, both in [27] and [34], without relying on an automata-based approach, however with no complexity result about the constrained NE existence problem. In case of games with ω-recognizable preference relations, our proof that NEs always exist relies on the technique developed in [27].

The results we obtain with games with ω-recognizable preference relations cover a large part of the games studied classically. In addition, our setting allows any combinations of objectives as soon as they are expressible by automata. However, it does not cover games with mean-payoff or energy objectives. Indeed, in the first case, it is proved in [2] that the related preference relation is not ω-automatic; and in the second case, the constrained NE existence problem is undecidable [11]. Note that the general concepts of ω-automatic and ω-recognizable relations have also been used to study imperfect information in games in [4, 8] and formal verification of quantitative systems in [2].

[2] important classes of objectives such as Büchi, co-Büchi, reachability, and safety require at most three priorities

2 Preliminaries

In this section, we introduce the useful definitions of games with ω-automatic preference relations and give several illustrative examples.

Automatic Relations. Let Σ be a fixed finite alphabet. We consider *binary relations* $R \subseteq \Sigma^\omega \times \Sigma^\omega$ on infinite words over Σ. The relation R is *ω-automatic* if it is accepted by a deterministic finite parity automaton over the alphabet $\Sigma \times \Sigma$, that is, R is an ω-regular language over $\Sigma \times \Sigma$. The automaton reads pairs of letters by advancing synchronously on the two words. This behavior is illustrated in Figure 1 below. A relation R is *ω-recognizable* if it is equal to $\cup_{i=1}^{\ell} X_i \times Y_i$ where $X_i, Y_i \subseteq \Sigma^\omega$ are ω-regular languages over Σ [35]. Any ω-recognizable relation is ω-automatic [40].

We suppose that the reader is familiar with the usual notion of deterministic parity automaton (DPA) used to accept ω-automatic relations [35]. A run is accepting if the maximum priority seen infinitely often is even. In this paper, we also use other classical notions of automata: deterministic Büchi automata (DBA) and Rabin automata. See, e.g., [26] for general definitions, or [1, 32] for deeper details. We also need the concept of *generalized* parity automaton which is an automaton with a positive[3] Boolean combination of parity conditions. Given an automaton $\mathcal{A}$, its *size* $|\mathcal{A}|$ is its number of states.

Games with Preference Relations. An *arena* is a tuple $A = (V, E, \mathcal{P}, (V_i)_{i\in\mathcal{P}})$ where V is a finite set of *vertices*, $E \subseteq V \times V$ is a set of *edges*, $\mathcal{P}$ is a finite set of *players*, and $(V_i)_{i\in\mathcal{P}}$ is a partition of V, where V_i is the set of vertices *owned* by player i. We assume, w.l.o.g., that each $v \in V$ has at least one *successor*, i.e., there exists $v' \in V$ such that $(v, v') \in E$. We define a *play* $\pi \in V^\omega$ (resp. a *history* $h \in V^*$) as an infinite (resp. finite) sequence of vertices $\pi_0\pi_1\dots$ such that $(\pi_k, \pi_{k+1}) \in E$ for any two consecutive vertices π_k, π_{k+1}. The set of all plays of an arena A is denoted $\mathsf{Plays}_A \subseteq V^\omega$, and we write Plays when the context is clear. The *length* $|h|$ of a history h is the number of its vertices. The empty history is denoted ε.

A *game* $\mathcal{G} = (A, (R_i)_{i\in\mathcal{P}})$ is an arena equipped with ω-automatic relations R_i over the alphabet V, one for each player i, called his *preference relation*. For any two plays π, π', player i *prefers* π' to π if $(\pi, \pi') \in R_i$. In the sequel, we write $\prec_i$ instead of R_i and for all $x, y \in V^\omega$, $x \prec_i y$, or $y \succ_i x$, instead of $(x, y) \in R_i$. We also say that x is *maximal* (resp. *minimal*) for $\prec_i$ if for all $y \in V^\omega$, we have $x \not\prec_i y$ (resp. $y \not\prec_i x$). Below we give various examples of games whose preference relations are all strict partial orders. At this stage, $\prec_i$ is just an ω-automatic relation without any additional hypotheses. Such hypotheses will be discussed in Section 6.

Given a play π and an index k, we write $\pi_{\geq k}$ the suffix $\pi_k\pi_{k+1}\dots$ of π. We denote the first vertex of π by $\mathsf{first}(\pi)$. These notations are naturally adapted to histories. We also write $\mathsf{last}(h)$ for the last vertex of a history $h \neq \varepsilon$. We can *concatenate* two non-empty histories h_1 and h_2 into a single one, denoted $h_1 \cdot h_2$ or h_1h_2 if $(\mathsf{last}(h_1), \mathsf{first}(h_2)) \in E$. When a history can be concatenated to itself, we call it *cycle*. A play $\pi = \mu\nu\nu\dots = \mu(\nu)^\omega$, where $\mu\nu$ is a history and ν a cycle, is called a *lasso*. The *length* of π is then the length of $\mu\nu$, denoted $|\pi|$.[4]

Let A be an arena. A *strategy* $\sigma_i : V^*V_i \to V$ for player i maps any history $h \in V^*V_i$ to a successor v of $\mathsf{last}(h)$, which is the next vertex that player i chooses to move after reaching the last vertex in h. A play $\pi = \pi_0\pi_1\dots$ is *consistent* with σ_i if $\pi_{k+1} = \sigma_i(\pi_0 \dots \pi_k)$ for all

[3] The negation is not allowed in the Boolean combination.
[4] To have a well-defined length for a lasso π, we assume that $\pi = \mu(\nu)^\omega$ with $\mu\nu$ of minimal length.

$k \in \mathbb{N}$ such that $\pi_k \in V_i$. Consistency is naturally extended to histories. A tuple of strategies $\sigma = (\sigma_i)_{i \in \mathcal{P}}$ with σ_i a strategy for player i, is called a *strategy profile*. The play π starting from an initial vertex v_0 and consistent with each σ_i is denoted by $\langle \sigma \rangle_{v_0}$ and called *outcome*.

A strategy σ_i for player i is *finite-memory* [26] if it can be encoded by a *Mealy machine* $\mathcal{M} = (M, m_0, \alpha_U, \alpha_N)$ where M is the finite set of memory states, $m_0 \in M$ is the initial memory state, $\alpha_U : M \times V \to M$ is the update function, and $\alpha_N : M \times V_i \to V$ is the next-move function. Such a machine defines the strategy σ_i such that $\sigma_i(hv) = \alpha_N(\widehat{\alpha}_U(m_0, h), v)$ for all histories $hv \in V^* V_i$, where $\widehat{\alpha}_U$ extends α_U to histories as expected. A strategy σ_i is *memoryless* if it is encoded by a Mealy machine with only one state.

We suppose that the reader is familiar with the concepts of two-player *zero-sum* games with ω-regular objectives and of winning strategy [21, 26].

Generality of the ω-Automatic Preference Framework. Let us show that the above notion of game $\mathcal{G} = (A, (\prec_i)_{i \in \mathcal{P}})$ encompasses many cases of classic games and more. We begin with games where each player i has an ω-regular objective $\Omega_i \subseteq V^\omega$, such as a reachability or a Büchi objective [21, 26]. In this case, the preference relation $\prec_i \subseteq V^\omega \times V^\omega$ is defined by $x \prec_i y$ if and only if $\Omega_i(x) < \Omega_i(y)$, where Ω_i is seen as a function $\Omega_i : V^\omega \to \{0, 1\}$. As Ω_i is ω-regular, it follows that $\prec_i$ is ω-automatic. For instance, given a target set $T \subseteq V$, the first DPA of Figure 1 accepts $\prec_i$ when Ω_i is a *reachability* objective $\{x = x_0 x_1 \ldots \in V^\omega \mid \exists k, x_k \in T\}$; the second DPA accepts $\prec_i$ when Ω_i is a *Büchi* objective $\{x \in V^\omega \mid \mathsf{Inf}(x) \cap T \neq \varnothing\}$, where $\mathsf{Inf}(x)$ is the set of vertices seen infinitely many times in x.

More general preference relations can be defined from several ω-regular objectives $(\Omega_i^j)_{1 \leq j \leq n}$ for player i. With each $x \in V^\omega$ is associated the payoff vector $\bar{\Omega}_i(x) = (\Omega_i^1(x), \ldots, \Omega_i^n(x)) \in \{0, 1\}^n$. Given a strict partial order $<$ on these payoff vectors, we define a preference relation $\prec_i$ such that $x \prec_i y$ if and only if $\bar{\Omega}_i(x) < \bar{\Omega}_i(y)$ [7]. There exist several strict partial orders on the payoff vectors, like, for example, the lexicographic order, or the counting order, i.e., $\bar{\Omega}_i(x) < \bar{\Omega}_i(y)$ if and only if $|\{j \mid \Omega_i^j(x) = 1\}| < |\{j \mid \Omega_i^j(y) = 1\}|$. One can check that all preference relations studied in [7] are ω-automatic.

Let us move on to classical quantitative objectives, such as quantitative reachability, limsup or discounted-sum objectives [21, 26]. In this case, an objective for player i is now a function $\Omega_i : V^\omega \to \mathbb{Q} \cup \{\pm\infty\}$.[5] We then define a preference relation $\prec_i$ such that $x \prec_i y$ if and only if $\Omega_i(x) < \Omega_i(y)$. Bansal et al. showed in [2] that such a relation is ω-automatic for a limsup objective and for a discounted-sum objective with an integer discount factor. They also proved that $\prec_i$ is not ω-automatic for a mean-payoff objective. The first DPA of Figure 1 where the label on the loop on the vertex with priority 0 is replaced by $(*, *)$, accepts a preference relation $\prec_i$ defined from a *min-cost-reachability* objective as follows: $x \prec_i y$ if and only if there exists ℓ such that $y_\ell \in T$ and, for all k, $x_k \in T \Rightarrow \exists \ell < k,\ y_\ell \in T$ (player i prefers plays with fewer steps to reach the target set T). The variant where player i prefers to maximize the number of steps to reach T,[6] called *max-reward-reachability*, is accepted by the third DPA in Figure 1.

Hence, there are many ways to envision ω-automatic relations. Note that in our framework, the preference relations $\prec_i$ of a game $\mathcal{G}$ can vary from one player to another, where each relation $\prec_i$ can be defined from a combination of several objectives (see Example 1 below).

[5] It can also be a function $\Omega : E^\omega \to \mathbb{Q} \cup \{\pm\infty\}$.
[6] as each step corresponds to a reward.

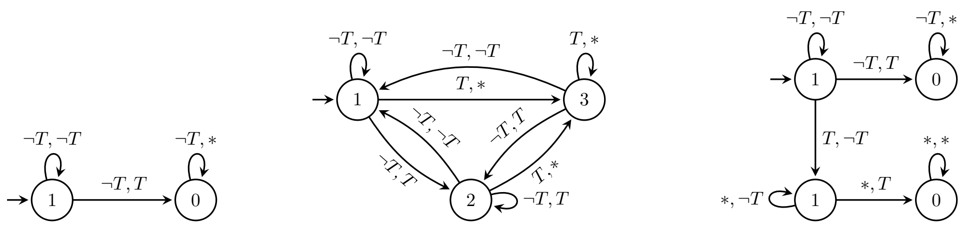

Figure 1 DPAs accepting preference relations, corresponding respectively to reachability, Büchi, and max-reward-reachability objectives. The priorities are indicated inside each state, and an edge label T, $\neg T$, or $*$ means that there is an edge for each label $v \in T$, $v \in V \setminus T$, and $v \in V$, respectively.

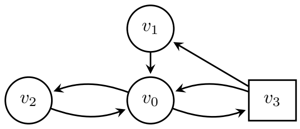

Figure 2 An arena with round (resp. square) vertices owned by player 1 (resp. player 2).

3 Decision Problems about Nash Equilibria

In this section, we state the decision problems studied in this paper and we provide our main results regarding their complexity classes.

Studied Problems. A *Nash Equilibrium* (NE) from an initial vertex v_0 is a strategy profile $(\sigma_i)_{i \in \mathcal{P}}$ such that for all players i and all strategies τ_i of player i, we have $\langle \sigma \rangle_{v_0} \not\prec_i \langle \tau_i, \sigma_{-i} \rangle_{v_0}$, where σ_{-i} denotes $(\sigma_j)_{j \in \mathcal{P} \setminus \{i\}}$. So, NEs are strategy profiles where no single-player has an incentive to unilaterally deviate from his strategy. When there exists a strategy τ_i such that $\langle \sigma \rangle_{v_0} \prec_i \langle \tau_i, \sigma_{-i} \rangle_{v_0}$, we say that τ_i (or, by notation abuse, $\langle \tau_i, \sigma_{-i} \rangle_{v_0}$) is a *profitable deviation* for player i. The set of players $\mathcal{P} \setminus \{i\}$ is called *coalition* $-i$, sometimes seen as one player opposed to player i.

▶ **Example 1.** Let us illustrate the NE definition with two examples. We consider the two-player arena depicted in Figure 2 such that player 2 owns only v_3 and player 1 owns all other vertices. The preference relation $\prec_1$ for player 1 is defined from a min-cost-reachability objective with a target set $T_1 = \{v_1\}$. The preference relation $\prec_2$ for player 2 is defined from a Büchi objective with a target set $T_2 = \{v_2\}$. Let us consider the strategy profile $\sigma = (\sigma_1, \sigma_2)$ defined by two memoryless strategies such that $\sigma_1(v_0) = v_3$ and $\sigma_2(v_3) = v_0$.[7] It is an NE from the initial vertex v_0 with outcome $\langle \sigma \rangle_{v_0} = (v_0 v_3)^\omega$. Player 1 has no profitable deviation if player 2 sticks on his strategy σ_2: it is not possible to visit vertex v_1. Player 2 has also no profitable deviation. There exists another NE $\sigma' = (\sigma'_1, \sigma'_2)$ from v_0 such that

- $\sigma'_1(hv_0) = v_2$ if the history h visits v_1, and to $\sigma'_1(hv_0) = v_3$ otherwise,
- σ'_2 is the memoryless strategy such that $\sigma'_2(v_3) = v_1$.

In that case, the NE outcome is $\langle \sigma' \rangle_{v_0} = v_0 v_3 v_1 (v_0 v_2)^\omega$. Note that both players prefer the second NE as $\langle \sigma \rangle_{v_0} \prec_i \langle \sigma' \rangle_{v_0}$ for $i = 1, 2$.

Let us slightly modify the relation of player 1 such that $\prec_1$ is defined from a lexicographic order using two objectives: a min-cost-reachability objective Ω_1^1 with T_1 and a Büchi objective Ω_1^2 with T_2. We have $x \prec_1 y$ if and only if $(\Omega_1^1(x) < \Omega_1^1(y))$ or $(\Omega_1^1(x) = \Omega_1^1(y)$ and

7 As v_1 and v_2 have only one successor, the strategy is trivially defined for those vertices.

$\Omega_1^2(x) < \Omega_1^2(y)$). If we consider the two previous strategy profiles, σ' is still an NE, but σ is no longer an NE as player 1 has a profitable deviation. Indeed, with the memoryless strategy τ_1 such that $\tau_1(v_0) = v_2$, we get $\langle\sigma\rangle_{v_0} = (v_0v_3)^\omega \prec_1 \langle\tau_1, \sigma_2\rangle_{v_0} = (v_0v_2)^\omega$. ⌟

▶ **Example 2.** In this example, we show that there does not always exist an NE in games with ω-automatic preference relations. Consider the simple one-player game $\mathcal{G}$ with two vertices v_0, v_1, the edges $(v_0, v_0), (v_0, v_1), (v_1, v_1)$, and whose preference relation $\prec_1$ is defined from a max-reward-reachability objective with a target set $T = \{v_1\}$. This game has no NE from the initial vertex v_0.[8] Indeed, if the strategy of player 1 is to loop on v_0, then he has a profitable deviation by going to T at some point, and if his strategy is to loop k times in v_0 and then go to T, then he has a profitable deviation by looping one more time in v_0 before going to T. ⌟

In this paper, we investigate the following problems.

▶ **Problems.**

- *The* NE checking problem *is to decide, given a game $\mathcal{G}$, an initial vertex v_0, and a strategy profile $\sigma = (\sigma_i)_{i\in\mathcal{P}}$ where each strategy σ_i is defined by a Mealy machine $\mathcal{M}_i$, whether σ is an NE from v_0 in $\mathcal{G}$.*
- *The* NE outcome checking problem *is to decide, given a game $\mathcal{G}$ and a lasso π, whether π is the outcome of an NE in $\mathcal{G}$.*
- *The* NE existence problem *is to decide, given a game $\mathcal{G}$ and an initial vertex v_0, whether there exists an NE from v_0 in $\mathcal{G}$.*
- *The* constrained NE existence problem *is to decide, given a game $\mathcal{G}$, an initial vertex v_0, and a lasso π_i for each player i, whether there exists an NE from v_0 in $\mathcal{G}$ with an outcome ρ such that $\pi_i \prec_i \rho$ for all players $i \in \mathcal{P}$.*

Main Results. Let us state our main results. We consider games $\mathcal{G} = (A, (\prec_i)_{i\in\mathcal{P}})$ on the arena $A = (V, E, \mathcal{P}, (V_i)_{i\in\mathcal{P}})$, where each preference relation $\prec_i \subseteq V^\omega \times V^\omega$ is ω-automatic. We denote by $\mathcal{A}_i$ the DPA accepting $\prec_i$ and by $\{0, 1, \dots, d_i\}$ its set of priorities. We say that a problem L is Parity-hard if there exists a polynomial reduction from the problem of deciding the winner of a two-player zero-sum parity game to L.

▶ **Theorem 3.** *The NE checking problem is* PSPACE*-complete.*

▶ **Theorem 4.** *The NE outcome checking problem is in* NP ∩ coNP *and* Parity*-hard.*

▶ **Theorem 5.** *The NE existence problem is exponential in $|V|$, $\Pi_{i\in\mathcal{P}}|\mathcal{A}_i|$, and $\Sigma_{i\in\mathcal{P}}d_i$, thus doubly exponential in $|\mathcal{P}|$. If the number of players is fixed (resp. for a one-player game), this problem is in* EXPTIME *and* PSPACE*-hard (resp.* PSPACE*-complete).*

▶ **Theorem 6.** *The constrained NE existence problem, with the constraints given by lassoes $(\pi_i)_{i\in\mathcal{P}}$, is exponential in $|V|$, $\Pi_{i\in\mathcal{P}}|\mathcal{A}_i|$, $\Pi_{i\in\mathcal{P}}|\pi_i|$, and doubly exponential in $\Sigma_{i\in\mathcal{P}}d_i$, thus also doubly exponential in $|\mathcal{P}|$. If the number of players and each d_i are fixed (resp. for a one-player game), this problem is in* EXPTIME *and* PSPACE*-hard (resp.* PSPACE*-complete).*

The proofs of these theorems are detailed in the next two sections. In Section 7, we reconsider the studied problems in the special case of games with ω-*recognizable* relations.

[8] A similar example is given in [34].

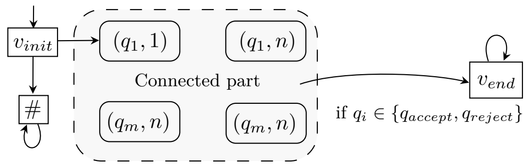

Figure 3 The game used for PSPACE-hardness of Theorem 3.

4 NE Checking and NE Outcome Checking Problems

We first prove Theorem 3, stating the PSPACE-completeness of the NE checking problem. The hardness is limited to a sketch of proof, the full technical details are given in the long version of this paper [15].

Proof of Theorem 3. We begin with the membership result. Given the Mealy machines $\mathcal{M}_i = (M_i, m_0^j, \alpha_U^i, \alpha_N^i)$, $i \in \mathcal{P}$, and the strategies σ_i they define, we have to check whether $\sigma = (\sigma_i)_{i \in \mathcal{P}}$ is an NE from a given initial vertex v_0. Equivalently, we have to check whether there exists a strategy τ_i for some player i such that $\langle \sigma \rangle_{v_0} \prec_i \langle \tau_i, \sigma_{-i} \rangle_{v_0}$ (in which case σ is not an NE). That is, whether there exists i such that the language

$$L_i = \{(x, y) \in V^\omega \times V^\omega \mid x \prec_i y,\ x = \langle \sigma \rangle_{v_0},\ y \text{ consistent with } \sigma_{-i} \text{ and starting at } v_0\}$$

is non-empty. We are going to describe a generalized DPA $\mathcal{B}_i$, with a conjunction of three parity conditions, that accepts L_i. We proceed as follows.

1. The set $\{(x, y) \in V^\omega \times V^\omega \mid x \prec_i y\}$ is accepted by the given DPA $\mathcal{A}_i$ that accepts $\prec_i$.
2. The outcome $\langle \sigma \rangle_{v_0}$ is a lasso obtained from the product of the arena A and all $\mathcal{M}_j$. We can define a DPA, of size exponential in the number of players, that only accepts $\langle \sigma \rangle_{v_0}$.
3. Finally, consider the product A' of the arena A with all $\mathcal{M}_j$, with $j \neq i$. We denote by V' the set of vertices of A', where each vertex is of the form $(v, (m_j)_{j \neq i})$, with $v \in V$ and m_j a memory state of $\mathcal{M}_j$. The set of plays y consistent with σ_{-i} and starting at v_0 is accepted by a DPA whose set of states is $V' \cup \{s_0\}$ with s_0, a new state, its initial state, all those states with priority 0, and whose transition function δ is such that $\delta((v, (m_j)_{j \neq i}), v') = (v', (m'_j)_{j \neq i})$ for $\alpha_U^j(m_j, v) = m'_j$, and $\delta(s_0, v_0) = (v_0, (m_0^j)_{j \neq i})$. Note that δ is a function as each $\mathcal{M}_j$ is deterministic and that this DPA is of exponential size in the number of players.

The announced automaton $\mathcal{B}_i$ is the product of the automata defined in the previous steps. It has exponential size and can be constructed on the fly, hence leading to a PSPACE algorithm. Indeed, to check whether L_i is non-empty, we guess a lasso $\mu(\nu)^\omega$ and its exponential length, and check whether the guessed lasso is accepted by $\mathcal{B}_i$. This only requires a polynomial space as the lasso is guessed on the fly, state by state, while computing the maximum priority occurring in ν for each priority function, and the length $|\mu\nu|$ is stored in binary. Finally, we repeat this procedure for each automaton $\mathcal{B}_i$, $i \in \mathcal{P}$.

For the PSPACE-hardness, we use a reduction from the membership problem for linear bounded deterministic Turing machines (LBTMs), known to be PSPACE-complete [24], to the complement of the NE checking problem. Recall that an LBTM T has a limited memory such that the tape head must remain in the n cells that contain the input word w.

We give only a sketch of proof. First, let us show how we encode any configuration of the LBTM. For the current word written on the tape, we associate one player per cell, and we say that the letter in the i-th cell, $i \in \{1, \ldots, n\}$, is the current memory state of the Mealy machine $\mathcal{M}_i$ of player i. Then we define an arena where each vertex is of the form (q, i), for a state q of T and the current position i of the tape head, and such that player i owns all the vertices (q, i). Second, we simulate transitions of the LBTM with the Mealy machines: $\mathcal{M}_i$ can describe the next vertex according to its memory state. For example, from vertex (q, i) and memory state a for player i, $\mathcal{M}_i$ moves to vertex $(q', i+1)$ and updates its memory state to a' if the LBTM says that from state q and letter a, the tape head must write a' and go right, and that the next state is q'.

This construction allows us to completely simulate the LBTM with an arena, described in Figure 3. We add an extra player $n+1$ who decides whether to let the other players follow their Mealy machine to simulate the LBTM on the given word, or go to a sink state #. With his preference relation $\prec_{n+1}$, player $n+1$ prefers a play visiting a vertex (q_{accept}, i), for any i, to any other play. His Mealy machine goes from v_{init} to #. Thus, the strategy profile given by all Mealy machines is not an NE if and only if it is profitable for player $n+1$ to let the other players simulate the LBTM on w, i.e., this simulation visits q_{accept}. ◀

Let us now prove Theorem 4 stating the complexity of the NE outcome checking problem.

Proof of Theorem 4. Let us begin with the membership result. Given a lasso π starting at v_0, checking whether π is an NE outcome amounts to finding a strategy profile $\sigma = (\sigma_i)_{i\in\mathcal{P}}$ with outcome π such that for all $i \in \mathcal{P}$ and all strategies τ_i, we have $\pi \not\prec_i \langle\tau_i, \sigma_{-i}\rangle_{v_0}$. In other words, given σ a strategy profile partially defined such that $\pi = \langle\sigma\rangle_{v_0}$, our goal is to check whether, for all i, there exists σ_{-i} that extends this partially defined profile such that for all τ_i, $\pi \not\prec_i \langle\tau_i, \sigma_{-i}\rangle_{v_0}$. For this purpose, we explain the algorithm in $\mathsf{NP} \cap \mathsf{coNP}$ for one given player $i \in \mathcal{P}$, and then repeat it for the other players.

Let us consider $L_i = \{x \in V^\omega \mid \pi \not\prec_i x\}$. This set is accepted by a DPA $\mathcal{B}_i$ constructed as the product of the complement of $\mathcal{A}_i$ and the lasso π. Clearly, the size of $\mathcal{B}_i$ is polynomially bounded in the sizes of $\mathcal{A}_i$ and π. So, L_i contains all the deviations that are not profitable for player i compared to π. Now, it suffices to decide whether the coalition $-i$ has a strategy σ_{-i} against player i to ensure that every play consistent with σ_{-i} lies in L_i. As L_i is accepted by the DPA $\mathcal{B}_i$, this amounts to solving a zero-sum parity game $\mathcal{H}_i$ (of polynomial size) defined directly from $\mathcal{B}_i$. The details are as follows.

Suppose that $\mathcal{B}_i$ has a set Q of states, an initial state q_0, and a transition function $\delta_{\mathcal{B}_i} : Q \times V \to Q$. Let us define the game $\mathcal{H}_i$, where the two players are A and B. Its set of vertices is the Cartesian product $V \times Q$, such that player A (resp. player B) controls the vertices (v, q) with $v \in V_i$ (resp. $v \notin V_i$). In other words, A has the role of player i while B has the role of the coalition $-i$. As $\mathcal{B}_i$ is deterministic, it is seen as an observer, and its states are information added to the vertices of V. Hence, the edges of $\mathcal{H}_i$ are of the form $((v, q),(v', q'))$ such that $(v, v') \in E$ and $q' = \delta_{\mathcal{B}_i}(q, v)$. We define a parity objective for player B as follows: the priority of each vertex (v, q) of $\mathcal{H}_i$ is equal to the priority of q in $\mathcal{B}_i$. Consequently, a play in $\mathcal{H}_i$ is won by player B if and only if the projection on its first component belongs to L_i. For the constructed game $\mathcal{H}_i$, from every vertex (v, q), we can decide in $\mathsf{NP} \cap \mathsf{coNP}$ which player wins in $\mathcal{H}_i$ together with a memoryless winning strategy for that player [26].

To obtain an algorithm in NP, it remains to check whether π, seen as a lasso in $\mathcal{H}_i$, only crosses vertices (v, q) that are winning for player B whenever $v \in V_i$. Indeed, in this case, we can deduce from a winning strategy τ_B from (v, q) for player B, a strategy σ_{-i} for the coalition $-i$ such that for all τ_i, $\pi \not\prec_i \langle\tau_i, \sigma_{-i}\rangle_{v_0}$. Similarly, to obtain an algorithm in coNP, we check whether π in $\mathcal{H}_i$ crosses at least one vertex (v, q) that is winning for player A and deduce a winning strategy for player i.

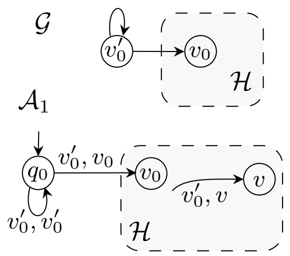

Figure 4 The game $\mathcal{G}$ and the DPA $\mathcal{A}_1$ accepting $\prec_1$ for the reduction of Theorem 4.

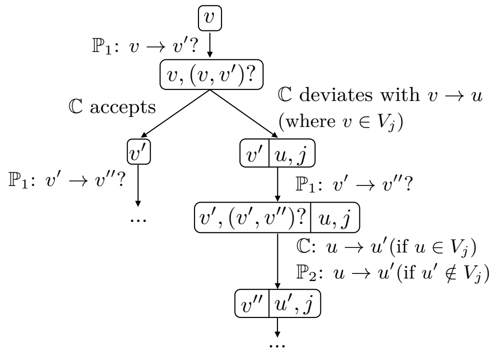

Figure 5 An illustration of the $\mathbb{P}_1\mathbb{C}\mathbb{P}_2$ game intuition: $\mathbb{P}_1$ observes the left part of a vertex (all v, v', v''), while $\mathbb{C}$ and $\mathbb{P}_2$ represent the deviating player j and the coalition $-j$ in the right part (all u, u'). Given $\rho = vv'v'' \dots$ and $\rho' = vuu' \dots$, $\mathbb{P}_1$ and $\mathbb{P}_2$ aim to ensure $\rho \not\prec_j \rho'$.

We continue with the hardness result, with a reduction from the problem of deciding whether player 1 has a winning strategy in a zero-sum parity game. We reduce this problem to the complement of the NE outcome checking problem, to establish its parity-hardness. Let $\mathcal{H}$ be a parity game with players 1 and 2, an arena A with V as set of vertices, an initial vertex v_0, and a priority function $\alpha : V \to \{0, \dots, d\}$. We construct a new game $\mathcal{G} = (A', \prec_1, \prec_2)$ with the same players, whose arena A' is a copy of A with an additional vertex v_0' owned by player 1, with v_0 and itself as successors (see Figure 4). Given $V' = V \cup \{v_0'\}$, the preference relation $\prec_2$ is empty, accepted by a one-state DPA, while the preference relation $\prec_1$ is defined as follows: $x \prec_1 y$ if and only if $x = (v_0')^\omega$ and $y = (v_0')^m y'$, with $m \geq 0$ and y' is a play in $\mathcal{H}$ starting at v_0 and satisfying the parity condition α. A DPA $\mathcal{A}_1$ accepting $\prec_1$ is depicted in Figure 4, it is constructed with a copy of the arena A and a new state q_0 with priority 1.

The proposed reduction is correct. Indeed, suppose that $\pi = (v_0')^\omega$ is not an NE outcome. As $\prec_2$ is empty, there cannot be profitable deviations for player 2. This means that for each strategy profile $\sigma = (\sigma_1, \sigma_2)$ with outcome π, there exists a deviating strategy τ_1 of player 1 such that $\pi \prec_1 \rho$ with $\rho = \langle \tau_1, \sigma_2 \rangle_{v_0'}$. Thus, by definition of $\prec_1$, ρ is equal to $(v_0')^m \rho'$ with ρ' a winning play in $\mathcal{H}$. Hence, transferred to $\mathcal{H}$, we get that for each strategy σ_2' of player 2, there exists a strategy τ_1' of player 1 such that $\langle \tau_1', \sigma_2' \rangle_{v_0}$ is winning. By determinacy, player 1 has thus a winning strategy in $\mathcal{H}$ from v_0. The other direction is proved similarly, if player 1 has a winning strategy in $\mathcal{H}$, then transferring this strategy to $\mathcal{G}$ gives a profitable deviation from a strategy with outcome π. ◀

5 NE Existence and Constrained NE Existence Problems

This section is devoted to the NE existence problem and its constrained variant. We mainly focus on the NE existence problem and explain at the end of the section how to take into account the constraints imposed on the NE outcome.

To solve the NE existence problem, we adapt a recent approach proposed in [16]. The idea is to reduce our problem to solving a *three-player game with imperfect information.*[9] Let us first give some intuition (see also Figure 5) and then the formal definition. We use

[9] Although the underlying game structure of [16] is reused, the player roles and correctness arguments differ entirely.

a reduction to a game with three players: two *Provers* $\mathbb{P}_1$ and $\mathbb{P}_2$ and one *Challenger* $\mathbb{C}$. The two Provers aim to build an NE outcome ρ while Challenger contests that it is an NE outcome: $\mathbb{P}_1$ has the task of building ρ edge by edge, while $\mathbb{P}_2$ has the task of showing that the deviation ρ' of player i proposed by $\mathbb{C}$ is not profitable, i.e., $\rho \not\prec_i \rho'$. We need two Provers (we cannot use a two-player zero-sum game), as the construction of ρ cannot depend on one specific deviation and must be fixed, i.e., its construction cannot change according to the deviation ρ' to artificially force $\rho \not\prec_i \rho'$. This also means that $\mathbb{P}_1$ has to build ρ without knowing when $\mathbb{C}$ deviates: he has *partial observation* of the game, while $\mathbb{C}$ and $\mathbb{P}_2$ have perfect information. This game, called $\mathbb{P}_1\mathbb{C}\mathbb{P}_2$ *game*, is articulated in two parts. The first part consists of vertices where $\mathbb{C}$ does not deviate, where an action of $\mathbb{P}_1$ is to suggest an edge (v, v') to extend the current construction of ρ, and an action of $\mathbb{C}$ is either to accept it or to deviate from ρ by choosing another edge (v, u) with $u \neq v'$. Such a deviation corresponds to a deviation by the player j who owns v, leading to the second part of the game. In this part, the vertices must retain the construction of the play ρ, the construction of the deviation ρ', and the component j to identify the player who deviated: $\mathbb{P}_1$ continues to propose an extension (v, v') for ρ with no interaction with $\mathbb{C}$, and $\mathbb{C}$ and $\mathbb{P}_2$, representing respectively the deviating player j and the opposed coalition $-j$, interact to construct ρ'. When the game stays in the first part, the objective of $\mathbb{P}_1$ is to produce an NE outcome ρ, and if it goes in the second part, $\mathbb{P}_1$ has the same goal and the aim of $\mathbb{P}_2$ is to retaliate on the deviations proposed by $\mathbb{C}$ to guarantee that ρ' is not a profitable deviation. Hence, the vertices of the $\mathbb{P}_1\mathbb{C}\mathbb{P}_2$ game also store the current states of the DPAs accepting the preference relations, in a way to compare the outcome ρ with the deviation ρ'.

We now proceed to the formal definition of the $\mathbb{P}_1\mathbb{C}\mathbb{P}_2$ game. Suppose that we are given a game $\mathcal{G} = (A, (\prec_i)_{i\in\mathcal{P}})$ with $A = (V, E, \mathcal{P}, (V_i)_{i\in\mathcal{P}})$ and $v_0 \in V$ as the initial vertex, and each relation $\prec_i$ accepted by a DPA $\mathcal{A}_i$. We denote each automaton as $\mathcal{A}_i = (Q_i, q_i^0, V \times V, \delta_i, \alpha_i)$ with Q_i its set of states, q_i^0 its initial state, $V \times V$ its alphabet, $\delta_i : Q_i \times (V \times V) \to Q_i$ its transition function, and $\alpha_i : Q_i \to \{0, 1, \dots, d_i\}$ its priority function. The game

$$\mathbb{P}_1\mathbb{C}\mathbb{P}_2(\mathcal{G}) = (S, (S_{\mathbb{P}_1}, S_{\mathbb{C}}, S_{\mathbb{P}_2}), (A_{\mathbb{P}_1}, A_{\mathbb{C}}, A_{\mathbb{P}_2}), \Delta, Obs, W_{\mathbb{P}_1\mathbb{P}_2})$$

is a three-player game with partial observation for $\mathbb{P}_1$, defined as follows.

- The set S of vertices are of the form $(v, j, u, (q_i)_{i\in\mathcal{P}})$ or $(v, j, u, (q_i)_{i\in\mathcal{P}}, (v, v'))$ such that $v, u \in V$, $j \in \mathcal{P} \cup \{\perp\}$, $q_i \in Q_i$, and $(v, v') \in E$. Coming back to the intuition given above, v is the current vertex of ρ, j is the deviating player (or $\perp$ if $\mathbb{C}$ did not deviate yet), u is the current vertex of ρ' (if it exists, otherwise $u = v$), q_i is the current state of $\mathcal{A}_i$ while comparing ρ and ρ'.
  Given that we are looking for an NE in $\mathcal{G}$ from some initial vertex v_0, we consider the initial vertex $s_0 = (v_0, \perp, v_0, (q_i^0)_i)$ in the $\mathbb{P}_1\mathbb{C}\mathbb{P}_2$ game.
- The set S is partitioned as $S_{\mathbb{P}_1} \cup S_{\mathbb{C}} \cup S_{\mathbb{P}_2}$ such that $S_{\mathbb{P}_1}$ is composed of the vertices $(v, j, u, (q_i)_i)$, $S_{\mathbb{C}}$ is composed of the vertices $(v, j, u, (q_i)_i, (v, v'))$ such that either $j = \perp$ and $v = u$, or $j \neq \perp$ and $u \in V_j$, and $S_{\mathbb{P}_2}$ is composed of the vertices $(v, j, u, (q_i)_i, (v, v'))$ such that $j \neq \perp$ and $u \in V \setminus V_j$.
- The set of actions[10] is, respectively for each player, equal to: $A_{\mathbb{P}_1} = \{(v, v') \mid (v, v') \in E\}$ ($\mathbb{P}_1$ chooses an edge (v, v') to extend the current construction of ρ) and $A_{\mathbb{C}} = A_{\mathbb{P}_2} = V$ ($\mathbb{C}$ and $\mathbb{P}_2$ choose the next vertex u of ρ' in case $\mathbb{C}$ deviates, otherwise $\mathbb{C}$ accepts the vertex v' of the edge (v, v') proposed by $\mathbb{P}_1$.)

[10] We introduce actions in a way to easily define the transition function Δ.

- The transition function is defined as follows:
  - for $\mathbb{P}_1$: for each $s = (v, j, u, (q_i)_i) \in S_{\mathbb{P}_1}$ and each $(v, v') \in A_{\mathbb{P}_1}$, we have $\Delta(s, (v, v')) = (v, j, u, (q_i)_i, (v, v'))$,
  - for $\mathbb{C}$ who has not yet deviated: for each $s = (v, \bot, v, (q_i)_i, (v, v')) \in S_{\mathbb{C}}$ and each $u \in A_{\mathbb{C}}$ with $(v, u) \in E$, we have either $u = v'$ and $\Delta(s, u) = (v', \bot, v', (q'_i)_i)$ (which means that $\mathbb{C}$ accepts the edge proposed by $\mathbb{P}_1$), or $u \neq v'$, $v \in V_j$, and $\Delta(s, u) = (v', j, u, (q'_i)_i)$ (which means that $\mathbb{C}$ starts deviating), with $q'_i = \delta(q_i, (v, v))$, for each $i \in \mathcal{P}$, in both cases, i.e., the states of the DPAs are updated.
  - for $\mathbb{C}$ who has deviated and $\mathbb{P}_2$: for each $s = (v, j, u, (q_i)_i, (v, v'))$ and each u' with $(u, u') \in E$, we have either $u \in V_j$ and thus $s \in S_{\mathbb{C}}$, or $u \in V \backslash V_j$ and thus $s \in S_{\mathbb{P}_2}$, and in both cases, $\Delta(s, u') = (v', j, u', (q'_i)_i)$ with $q'_i = \delta(q_i, (v, u))$, for each $i \in \mathcal{P}$.
- The observation function *Obs* for $\mathbb{P}_1$[11] is such that $Obs((v, j, u, (q_i)_i, (v, v'))) = (v, v')$ and $Obs((v, j, u, (q_i)_i)) = v$. When $s, s' \in S$ and $Obs(s) = Obs(s')$, we consider that $\mathbb{P}_1$ cannot distinguish s and s'. Hence, $\mathbb{P}_1$ can only observe the vertices v of the initial game $\mathcal{G}$ and the edges (v, v') that he proposes. We naturally extend *Obs* to histories and plays of the $\mathbb{P}_1\mathbb{C}\mathbb{P}_2$ game by applying the observation function on each of their vertices.
- To complete the definition of the $\mathbb{P}_1\mathbb{C}\mathbb{P}_2$ game, it remains to define the winning condition $W_{\mathbb{P}_1\mathbb{P}_2}$. Let us introduce some notation. Given a vertex s, we denote by $\mathsf{proj}_{V,1}(s)$ (resp. $\mathsf{dev}(s)$, $\mathsf{proj}_{V,2}(s)$) the projection on its first (resp. second, third) component. For a vertex $s \in S_{\mathbb{C}} \cup S_{\mathbb{P}_2}$, we denote by $\mathsf{proj}_E(s)$ this last component of s. Note that if $s \in S_{\mathbb{P}_1}$, then $Obs(s) = \mathsf{proj}_{V,1}(s)$, and if $s \in S_{\mathbb{C}} \cup S_{\mathbb{P}_2}$, then $Obs(s) = \mathsf{proj}_E(s)$. Given a play $\pi = \pi_0\pi_1\pi_2\ldots$ of the $\mathbb{P}_1\mathbb{C}\mathbb{P}_2$ game starting at the initial vertex s_0, π is an alternation of vertices of $S_{\mathbb{P}_1}$ and vertices of $S_{\mathbb{C}} \cup S_{\mathbb{P}_2}$. Moreover, looking at the first (resp. third) components of the vertices of π, each such component is repeated from one vertex to the next one. Thus, we denote by $\mathsf{proj}_{V,1}(\pi)$ the projection on the first component of the vertices of $\pi_0\pi_2\ldots\pi_{2k}\ldots$. Similarly, we use notation $\mathsf{proj}_{V,2}(\pi)$ for the projection on the third component. We also define the notation $\mathsf{proj}_E(\pi)$ for the projection of $\pi_1\pi_3\ldots\pi_{2k+1}\ldots$ on the last component of its vertices. Note that $\mathsf{proj}_{V,1}(\pi) = Obs(\pi_0\pi_2\ldots\pi_{2k}\ldots)$ and $\mathsf{proj}_E(\pi) = Obs(\pi_1\pi_3\ldots\pi_{2k+1}\ldots)$. In the play π, either the second component always remains equal to $\bot$ or ultimately becomes equal to some $j \in \mathcal{P}$. We use notation $\mathsf{dev}(\pi)$ to denote this value $\bot$ or j. All these notations are also used for histories.

  The set $W_{\mathbb{P}_1\mathbb{P}_2}$ is defined as $W_{\mathbb{P}_1\mathbb{P}_2} = W_{acc} \cup W_{dev}$ where W_{acc} is the set of plays where $\mathbb{C}$ always agreed with $\mathbb{P}_1$ and W_{dev} is the set of plays where $\mathbb{C}$ deviated but $\mathbb{P}_2$ was able to show that this deviation is not profitable, i.e.,
  - $W_{acc} = \{\pi \in \mathsf{Plays}(\mathbb{P}_1\mathbb{C}\mathbb{P}_2(\mathcal{G})) \mid \mathsf{dev}(\pi) = \bot\}$,
  - $W_{dev} = \{\pi \in \mathsf{Plays}(\mathbb{P}_1\mathbb{C}\mathbb{P}_2(\mathcal{G})) \mid \exists j \in \mathcal{P}, \mathsf{dev}(\pi) = j \text{ and } \mathsf{proj}_{V,1}(\pi) \not\prec_j \mathsf{proj}_{V,2}(\pi)\}$.

  This set $W_{\mathbb{P}_1\mathbb{P}_2}$ is the winning condition for both $\mathbb{P}_1$ and $\mathbb{P}_2$ while $\mathbb{C}$ has the complementary winning condition $S^\omega \backslash W_{\mathbb{P}_1\mathbb{P}_2}$.

The next theorem states how the $\mathbb{P}_1\mathbb{C}\mathbb{P}_2$ game helps to solve the NE existence problem. A strategy $\tau_{\mathbb{P}_1}$ of $\mathbb{P}_1$ is *observation-based* if for all histories h, h' ending in a vertex of $\mathbb{P}_1$ such that $Obs(h) = Obs(h')$, we have $\tau_{\mathbb{P}_1}(h) = \tau_{\mathbb{P}_1}(h')$.

▶ **Theorem 7.** *The following statements are equivalent:*
- *In $\mathcal{G}$, there exists an NE $\sigma = (\sigma_i)_{i\in\mathcal{P}}$ from v_0,*
- *In $\mathbb{P}_1\mathbb{C}\mathbb{P}_2(\mathcal{G})$, there exists an observation-based strategy $\tau_{\mathbb{P}_1}$ of $\mathbb{P}_1$ such that for all strategies $\tau_{\mathbb{C}}$ of $\mathbb{C}$, there is a strategy $\tau_{\mathbb{P}_2}$ of $\mathbb{P}_2$ such that $\langle\tau_{\mathbb{P}_1}, \tau_{\mathbb{C}}, \tau_{\mathbb{P}_2}\rangle_{s_0} \in W_{\mathbb{P}_1\mathbb{P}_2}$.*

[11] Recall that $\mathbb{C}$ and $\mathbb{P}_2$ have total observation of the $\mathbb{P}_1\mathbb{C}\mathbb{P}_2$ game.

Theorem 7 is the key tool to solve the NE existence problem. It is proved in detail in the long version of this paper [15]. We give hereafter a sketch of proof for the membership result of Theorem 5, which follows the approach proposed in [16]. The PSPACE-hardness already holds for one-player games, with a reduction from the existence of a maximal element in a relation $\prec$, which is a PSPACE-complete problem and close to the existence of NEs in one-player games. All details are given in the long version [15]

Sketch of Proof of Theorem 5, Membership. By Theorem 7, deciding whether there exists an NE from v_0 in $\mathcal{G}$ reduces to deciding whether there exists an observation-based strategy $\tau_{\mathbb{P}_1}$ of $\mathbb{P}_1$ in $\mathbb{P}_1\mathbb{C}\mathbb{P}_2(\mathcal{G})$ such that for all strategies $\tau_{\mathbb{C}}$ of $\mathbb{C}$, there is a strategy $\tau_{\mathbb{P}_2}$ of $\mathbb{P}_2$ such that $\langle \tau_{\mathbb{P}_1}, \tau_{\mathbb{C}}, \tau_{\mathbb{P}_2}\rangle_{s_0} \in W_{\mathbb{P}_1\mathbb{P}_2}$. In [16], the authors solve the problem they study by solving a similar three-player game with imperfect information. They proceed at follows: *(i)* the winning condition is translated into a Rabin condition[12] on the arena of the $\mathbb{P}_1\mathbb{C}\mathbb{P}_2$ game, *(ii)* the three-player game is transformed into a two-player zero-sum Rabin game with imperfect information, and finally *(iii)* classical techniques to remove imperfect information are used to obtain a two-player zero-sum parity game with perfect information.

In this sketch of proof, we only explain the first step, i.e., how to translate $W_{\mathbb{P}_1\mathbb{P}_2} = W_{acc} \cup W_{dev}$ into a Rabin condition, as the second and third steps heavily use the arguments of [16]. To translate W_{acc}, we use one pair (E_1, F_1) such that $E_1 = \varnothing$ and $F_1 = \{s \in S \mid \mathsf{dev}(s) = \bot\}$. To translate W_{dev}, notice that $\mathsf{dev}(\pi) = j$ is equivalent to $\mathsf{dev}(\pi) \notin \{\bot\} \cup \mathcal{P}\backslash\{j\}$, and thus $W_{dev} = \cup_{j\in\mathcal{P}}\{\pi \in \mathsf{Plays}(\mathbb{P}_1\mathbb{C}\mathbb{P}_2(\mathcal{G})) \mid \mathsf{dev}(\pi) \notin \{\bot\} \cup \mathcal{P}\backslash\{j\}$ and $\mathsf{proj}_{V,1}(\pi) \not\prec_j \mathsf{proj}_{V,2}(\pi)\}$. Recall that each relation $\prec_j$ is accepted by the DPA $\mathcal{A}_j$ with the priority function $\alpha_j : Q_j \to \{0, 1, \dots d_j\}$, thus also $\not\prec_j$ with the modified priority function $\alpha_j + 1$. Therefore, W_{dev} can be translated into a Rabin condition on the vertices of S with $\Sigma_{j\in\mathcal{P}} d_j$ Rabin pairs [32]. Steps *(ii)* and *(iii)* are detailed in the long version [15], leading to the announced complexity: the NE existence problem is exponential in $|V|$, $\Pi_{i\in\mathcal{P}}|\mathcal{A}_i|$, and $\Sigma_{i\in\mathcal{P}} d_i$. ◀

Let us finally comment on Theorem 6 stating the complexity class of the constrained NE existence problem. The detailed proof is presented in the long version of this paper [15]. The approach to proving membership is very similar to that of the NE existence problem, as we only need to modify W_{acc} in a way to include the constraints imposed on the NE outcome. A constraint imposed by a lasso π_i can be represented by a DPA $\mathcal{A}'_i$ accepting the language $\{\rho \in V^\omega \mid \pi_i \prec_i \rho\}$, with a polynomial size $|\mathcal{A}_i| \cdot |\pi_i|$. Then it suffices to extend the arena the $\mathbb{P}_1\mathbb{C}\mathbb{P}_2$ game with the states of each $\mathcal{A}'_i$. The hardness result is obtained by a reduction from the NE existence problem, already for one-player games.

Note that by steps *(i)*-*(iii)*, solving the (constrained) NE existence problem is equivalent to solving a zero-sum parity game with memoryless winning strategies for both players. Therefore, we get the following property:

▶ **Corollary 8.** *If there exists a (constrained) NE, then there exists one with finite-memory strategies.*

There is a great interest in using the concept of $\mathbb{P}_1\mathbb{C}\mathbb{P}_2$ game, as it provides a unified approach to solve the NE existence problem and its constrained variant. With this approach, we could also decide the existence of an NE whose outcome ρ satisfies various combinations of constraints, such as, e.g., $\pi_i \prec_i \rho \prec_i \pi'_i$ for one or several players i. The chosen constraints only impact the winning condition $W_{\mathbb{P}_1\mathbb{P}_2}$ and thus its translation into a Rabin condition.

[12] Recall that a Rabin condition uses a finite set of pairs $(E_j, F_j)_{j\in J}$ in a way to accept plays π such that there exists $j \in J$ with $\mathsf{Inf}(\pi) \cap E_j = \varnothing$ and $\mathsf{Inf}(\pi) \cap F_j \neq \varnothing$.

6 Hypotheses on Preference Relations

In the previous sections, we presented several decision algorithms. Since the players' relations $\prec_i$ are intended to formalize how they prefer one play to another, we may naturally expect them to satisfy certain properties, such as *irreflexivity* and *transitivity*. However, since the relation $\prec_i$ is accepted by a DPA, its structure can be intricate, and it becomes relevant to verify whether such properties hold. In this section, we address decision problems related to the algorithmic verification of properties of $\prec_i$. We also explore alternative approaches to modeling preferences between plays, focusing in particular on cases where the DPA accepts a non-strict preference relation $\precsim_i$, i.e., a preorder, rather than a strict partial order.

Hypotheses on Relations. Given a relation $R \subseteq \Sigma^\omega \times \Sigma^\omega$, it is:

- *reflexive* (resp. *irreflexive*) if for all $x \in \Sigma^\omega$, we have $(x, x) \in R$ (resp. $(x, x) \notin R$),
- *transitive* (resp. $\neg$-*transitive*) if for all $x, y, z \in \Sigma^\omega$, we have $(x, y) \in R \wedge (y, z) \in R \Rightarrow (x, z) \in R$ (resp. $(x, y) \notin R \wedge (y, z) \notin R \Rightarrow (x, z) \notin R$),
- *total* if for all $x, y \in \Sigma^\omega$, we have $(x, y) \in R \vee (y, x) \in R$,

A reflexive and transitive relation is a *preorder*. An irreflexive and transitive relation is a *strict partial order*. When, in addition, a strict partial order R is $\neg$-transitive, it is a *strict weak order*. The next proposition states that all the relevant properties mentioned above can be *efficiently* verified on the DPA accepting the relation R. It is proved in the long version [15].

▶ **Proposition 9.** *The problem of deciding whether an ω-automatic relation R is reflexive (resp. irreflexive, transitive, $\neg$-transitive, total) is* NL*-complete.*

Variants on our Setting. Let us first observe that all the lower bounds established for the decision problems about NEs remain valid even when the players' preference relations R_i are assumed to be *strict partial orders*.[13] This implies that taking this additional property into account does not yield any important advantage in terms of complexity class.

We now consider an alternative setting in which each relation R_i is not a strict preference, but rather a *preorder* (where some plays are declared equivalent). To discuss this further, let us recall the relationship between strict partial orders and preorders. From a strict partial order $\prec$, one can obtain a preorder $\precsim$ by taking its reflexive closure, i.e., $x \precsim y$ if $x \prec y$ or $x = y$. When $\prec$ is a strict weak order, another preorder which is total, is obtained by defining $x \precsim y$ if $y \not\prec x$. In both cases, if $\prec$ is ω-automatic, $\precsim$ is also ω-automatic (resp. by a generalized DPA with a disjunction of two parity conditions, or by a DPA). Conversely, from any preorder $\precsim$, we can define a strict partial order $\prec$ such that $x \prec y$ if $x \precsim y \wedge y \not\precsim x$. Every strict partial order can be constructed this way. Moreover, if $\precsim$ is total, then $\prec$ is a strict weak order. We can also define the equivalence relation $\sim$ such that $x \sim y$ if $x \precsim y \wedge y \precsim x$. The equivalence class of x is denoted $[x]$. Again, if $\precsim$ is ω-automatic, then $\prec$ and $\sim$ are both ω-automatic (by a generalized DPA with a conjunction of two parity conditions).

For games with preorders R_i, we keep the same upper bounds, except that we only have an NP membership for Theorem 4. Indeed, it requires solving a generalized parity game with a disjunction of two parity conditions (instead of a parity game), solvable in NP [18]. All lower bounds remain valid by carefully modifying the preference relations used in the reductions into preorders, see the long version [15].

---

[13] The upper bounds do not need more than the ω-automaticity of each R_i

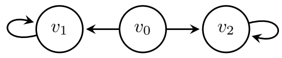

Figure 6 An arena with one player.

Finally, recall that given a lasso π, it is easy to construct an automaton that accepts all plays related to π according to an ω-automatic preference relation. This set being ω-regular, all standard verification techniques for ω-regular languages can be applied. For example, one may wish to verify that all plays preferred to π satisfy a given ω-regular property. In such cases, the full range of verification algorithms developed for ω-regular languages can be used.

Alternative Definition of NE. Given a game $\mathcal{G} = (A, (\precsim_i)_{i\in\mathcal{P}})$ with preorders $\precsim_i$, an NE is a strategy profile σ such that for all players i and all strategies τ_i of player i, we have $\langle\sigma\rangle_{v_0} \not\prec_i \langle\tau_i, \sigma_{-i}\rangle_{v_0}$. An alternative definition asks for all i and τ_i that $\langle\tau_i, \sigma_{-i}\rangle_{v_0} \precsim_i \langle\sigma\rangle_{v_0}$ [7]. The two definitions yield different notions of NE (unless all $\precsim_i$ are total). In this paper, we do not consider the second definition, due to the nonexistence of NEs in very simple games. Let us consider the one-player game $\mathcal{G}$ depicted in Figure 6, where, from the initial vertex v_0, player 1 has the choice between $v_0v_1^\omega$ and $v_0v_2^\omega$. We consider the preorder $\precsim_1$ equal to $\{(x,x) \mid x \in \{v_0, v_1, v_2\}^\omega\}$. Clearly, $v_0v_1^\omega \not\precsim_1 v_0v_2^\omega$ and $v_0v_2^\omega \not\precsim_1 v_0v_1^\omega$, showing that there is no NE from v_0 for this alternative definition (while $v_0v_1^\omega$ and $v_0v_2^\omega$ are both NE outcomes with the first definition). This phenomenon appears as soon as there are two incomparable plays.

7 ω-Recognizable Relations

In this section, we suppose that we have a game $\mathcal{G} = (A, (\precsim_i)_{i\in\mathcal{P}})$ whose relations $\precsim_i$ are *ω-recognizable* and *preorders*. We recall that $\precsim_i$ is ω-recognizable if it is of the form $\cup_{i=1}^{\ell} X_i \times Y_i$ where $X_i, Y_i \subseteq \Sigma^\omega$ are ω-regular languages over Σ. Any ω-recognizable relation is ω-automatic (see [40]), and deciding whether an ω-automatic relation accepted by a DPA is ω-recognizable is NL-complete [3]. For each $\precsim_i$, we use the related relations $\prec_i$ and $\sim_i$ as defined in the previous section.

In Example 2, we presented a one-player game with no NE. The reason for the absence of NE is that $\prec_1$ has an unbounded infinite ascending chain. This situation cannot happen for ω-recognizable preorders, as highlighted in the next proposition, easily derived from [35] (its proof is given in the long version [15]). This motivates the interest of games with ω-recognizable preference relations.

▶ **Proposition 10.** *An ω-automatic preorder $\precsim \subseteq \Sigma^\omega \times \Sigma^\omega$ is ω-recognizable if and only if its induced equivalence relation $\sim$ has finite index.*

Thanks to this result, we can partition Σ^ω as a *finite lattice* given by a partial order induced by $\precsim$ on the equivalence classes of $\sim$. In particular, there always exists a maximal (resp. minimal) element in this lattice. Examples of ω-recognizable preorders are numerous: those deriving from any Boolean combination of ω-regular objectives or any multidimensional objective where each dimension is defined using an ω-regular objective. In the subclass of games with ω-recognizable preorders, the main difference is the *existence of NEs*.

▶ **Theorem 11.** *When the preference relations of a game are all ω-recognizable preorders, then there always exists an NE composed of finite-memory strategies.*

The proof of Theorem 11 requires two steps. We first prove the existence of an NE under the assumption that each preference relation $\precsim_i$ is a *total* preorder and then without this assumption. (Note that we get an NE composed of finite-memory strategies by Corollary 8). The first step can be obtained as a corollary of [34, Theorem 15] that guarantees the existence of an NE in the case of strict weak orders $\prec_i$. Recall that the relation $\prec_i$ induced by a preorder $\precsim_i$ is a strict weak order if $\precsim_i$ is total. Nevertheless, we provide a proof of this first step in the long version of this paper [15], inspired by the work of [27] and [12], where the existence of NEs is studied through the concept of value and optimal strategy (see below and in the long version [15]).

The second step is obtained thanks to an embedding of partial preorders into total preorders, as described in the next proposition. Theorem 11 easily follows (see the long version [15]).

▶ **Proposition 12.** *Any ω-recognizable preorder $\precsim$ can be embedded into an ω-recognizable total preorder $\precsim'$. Moreover, for all x, y, if $x \bowtie y$, then $x \bowtie' y$, for $\bowtie \in \{\precsim, \prec, \succsim, \succ, \sim\}$.*

We now focus on *prefix-independent* relations R, such that for all $x, y \in \Sigma^\omega$, $(x, y) \in R \Leftrightarrow \forall u, v \in \Sigma^*, (ux, uy) \in R$. From our proof of Theorem 11, when the relations $\precsim_i$ are total and prefix-independent, we can derive the following *characterization* of NE outcomes in terms of values (the proof is given in the long version [15]). In this context, for each player i and vertex v of $\mathcal{G}$, there always exists a *value* $val_i(v)$ (which is an equivalence class of $\sim_i$) and *optimal strategies* σ_i^v for player i and σ_{-i}^v for the coalition $-i$ such that σ_i^v (resp. σ_{-i}^v) ensures consistent plays π starting at v such that $val_i(v) \precsim_i [\pi]_i$ (resp. $[\pi]_i \precsim_i val_i(v)$) (see the long version [15]). Such an NE characterization is well-known for games with classical objectives (see, e.g., the survey [13]).

▶ **Theorem 13.** *Let $\mathcal{G}$ be a game such that each preference relation $\precsim_i$ is an ω-recognizable preorder, total, and prefix-independent. Then a play $\rho = \rho_0\rho_1\dots$ is an NE outcome if and only if for all vertices ρ_n of ρ, if $\rho_n \in V_i$, then $val_i(\rho_n) \precsim_i [\rho]_i$.*

In this theorem, we can weaken the hypothesis of prefix-independency into prefix-linearity. A relation R is *prefix-linear* if, for all $x, y \in \Sigma^\omega$ and $u \in \Sigma^*$, $(x, y) \in R$ implies $(ux, uy) \in R$. In that case, the condition $val_i(\rho_n) \precsim_i [\rho]_i$ in Theorem 13 must be replaced by $val_i(\rho_n) \precsim_i [\rho_{\geq n}]_i$. Moreover, deciding whether a relation R is prefix-independent (resp. prefix-linear) is NL-complete. All proofs and details are provided in the long version [15].

8 Conclusion

In this work, we have introduced a general framework for defining players' preferences via ω-automatic preference relations instead of fixed reward functions. It subsumes several classical settings, including the Boolean setting with ω-regular objectives and quantitative models such as min-cost-reachability, as well as combinations of several such objectives.

In this framework, we have studied the complexity of four fundamental problems related to NEs, notably with a novel use of the $\mathbb{P}_1\mathbb{C}\mathbb{P}_2$ game setting recently introduced in [16]. This approach enables a broader applicability and more reusable results. It contrasts sharply with most existing work that is typically focused on specific reward functions.

We hope that our framework will serve as a basis for exploring additional problems such as decision problems about subgame perfect equilibria (which are NEs in any subgame of a game [38]), or the rational synthesis problem as studied in [33]. New results will lead to the development of general and modular solutions for a wider class of questions in the theory of infinite games played on graphs.

References

1 Christel Baier and Joost-Pieter Katoen. *Principles of model checking.* MIT Press, 2008.

2 Suguman Bansal, Swarat Chaudhuri, and Moshe Y. Vardi. Comparator automata in quantitative verification. *Logical Methods in Computer Science*, Volume 18, Issue 3, July 2022. doi:10.46298/lmcs-18(3:13)2022.

3 Pascal Bergsträßer and Moses Ganardi. Revisiting Membership Problems in Subclasses of Rational Relations. In *38th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2023, Boston, MA, USA, June 26-29, 2023*, pages 1–14. IEEE, 2023. doi:10.1109/LICS56636.2023.10175722.

4 Dietmar Berwanger and Laurent Doyen. Observation and Distinction: Representing Information in Infinite Games. *Theory of Computing Systems*, 67(1):4–27, 2023. doi:10.1007/s00224-021-10044-x.

5 Achim Blumensath and Erich Grädel. Automatic structures. In *15th Annual IEEE Symposium on Logic in Computer Science, Santa Barbara, California, USA, June 26-29, 2000*, pages 51–62. IEEE Computer Society, 2000. doi:10.1109/lics.2000.855755.

6 Patricia Bouyer, Romain Brenguier, and Nicolas Markey. Nash Equilibria for Reachability Objectives in Multi-player Timed Games. In Paul Gastin and François Laroussinie, editors, *CONCUR 2010 - Concurrency Theory, 21th International Conference, CONCUR 2010, Paris, France, August 31-September 3, 2010. Proceedings*, volume 6269 of *Lecture Notes in Computer Science*, pages 192–206. Springer, 2010. doi:10.1007/978-3-642-15375-4_14.

7 Patricia Bouyer, Romain Brenguier, Nicolas Markey, and Michael Ummels. Pure Nash Equilibria in Concurrent Deterministic Games. *Logical Methods in Computer Science*, 11(2), 2015. doi:10.2168/lmcs-11(2:9)2015.

8 Laura Bozzelli, Bastien Maubert, and Sophie Pinchinat. Uniform strategies, rational relations and jumping automata. *Information and Computation*, 242, 2015. doi:10.1016/j.ic.2015.03.012.

9 Romain Brenguier, Lorenzo Clemente, Paul Hunter, Guillermo A. Pérez, Mickael Randour, Jean-François Raskin, Ocan Sankur, and Mathieu Sassolas. Non-Zero Sum Games for Reactive Synthesis. In Adrian-Horia Dediu, Jan Janousek, Carlos Martín-Vide, and Bianca Truthe, editors, *Language and Automata Theory and Applications - 10th International Conference, LATA 2016, Prague, Czech Republic, March 14-18, 2016, Proceedings*, volume 9618 of *Lecture Notes in Computer Science*, pages 3–23. Springer, 2016. doi:10.1007/978-3-319-30000-9_1.

10 Léonard Brice, Jean-François Raskin, and Marie van den Bogaard. Rational Verification for Nash and Subgame-Perfect Equilibria in Graph Games. In Jérôme Leroux, Sylvain Lombardy, and David Peleg, editors, *48th International Symposium on Mathematical Foundations of Computer Science, MFCS 2023, August 28 to September 1, 2023, Bordeaux, France*, volume 272 of *LIPIcs*, pages 26:1–26:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/lipics.mfcs.2023.26.

11 Léonard Brice, Marie van den Bogaard, and Jean-François Raskin. Subgame-perfect Equilibria in Mean-payoff Games (journal version). *Logical Methods in Computer Science*, 19(4), 2023. doi:10.46298/lmcs-19(4:6)2023.

12 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.

13 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.

14 Véronique Bruyère. Synthesis of Equilibria in Infinite-Duration Games on Graphs. *ACM SIGLOG News*, 8(2):4–29, May 2021. doi:10.1145/3467001.3467003.

15 Véronique Bruyère, Christophe Grandmont, and Jean-François Raskin. Games with ω-Automatic Preference Relations. *CoRR*, abs/2503.04759, 2025. doi:10.48550/arXiv.2503.04759.

16 Véronique Bruyère, Jean-François Raskin, Alexis Reynouard, and Marie Van Den Bogaard. The Non-Cooperative Rational Synthesis Problem for Subgame Perfect Equilibria and omega-regular Objectives, 2024. doi:10.48550/arxiv.2412.08547.

17 Olivier Carton, Christian Choffrut, and Serge Grigorieff. Decision problems among the main subfamilies of rational relations. *RAIRO - Theoretical Informatics and Applications - Informatique Théorique et Applications*, 40(2):255–275, 2006. doi:10.1051/ita:2006005.

18 Krishnendu Chatterjee, Thomas A. Henzinger, and Nir Piterman. Generalized Parity Games. In Helmut Seidl, editor, *Foundations of Software Science and Computational Structures, 10th International Conference, FOSSACS 2007, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2007, Braga, Portugal, March 24-April 1, 2007, Proceedings*, volume 4423 of *Lecture Notes in Computer Science*, pages 153–167. Springer, 2007. doi:10.1007/978-3-540-71389-0_12.

19 Krishnendu Chatterjee and Vishwanath Raman. Synthesizing Protocols for Digital Contract Signing. In Viktor Kuncak and Andrey Rybalchenko, editors, *Verification, Model Checking, and Abstract Interpretation - 13th International Conference, VMCAI 2012, Philadelphia, PA, USA, January 22-24, 2012. Proceedings*, volume 7148 of *Lecture Notes in Computer Science*, pages 152–168. Springer, 2012. doi:10.1007/978-3-642-27940-9_11.

20 Yoav Feinstein, Orna Kupferman, and Noam Shenwald. Non-Zero-Sum Games with Multiple Weighted Objectives. In *Proceedings of the 28th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2025)*, volume To appear of *Lecture Notes in Computer Science*, page To appear. Springer, 2025. URL: https://www.cs.huji.ac.il/~ornak/publications/tacas25b.pdf.

21 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*. Online, 2023. doi:10.48550/arxiv.2305.10546.

22 János Flesch, Jeroen Kuipers, Ayala Mashiah-Yaakovi, Gijs Schoenmakers, Eilon Solan, and Koos Vrieze. Perfect-Information Games with Lower-Semicontinuous Payoffs. *Mathematics of Operations Research*, 35(4):742–755, 2010. doi:10.1287/moor.1100.0469.

23 Christiane Frougny and Jacques Sakarovitch. Synchronized rational relations of finite and infinite words. *Theoretical Computer Science*, 108(1):45–82, 1993. doi:10.1016/0304-3975(93)90230-Q.

24 M. R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979. URL: https://perso.limos.fr/~palafour/PAPERS/PDF/Garey-Johnson79.pdf.

25 Erich Grädel. Automatic Structures: Twenty Years Later. In Holger Hermanns, Lijun Zhang, Naoki Kobayashi, and Dale Miller, editors, *LICS '20: 35th Annual ACM/IEEE Symposium on Logic in Computer Science, Saarbrücken, Germany, July 8-11, 2020*, LICS'20, pages 21–34, New York, NY, USA, 2020. Association for Computing Machinery. doi:10.1145/3373718.3394734.

26 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.

27 Erich Grädel and Michael Ummels. Solution Concepts and Algorithms for Infinite Multiplayer Games. In *New Perspectives on Games and Interaction*, volume 4, pages 151–178. Amsterdam University Press, 2008. URL: https://core.ac.uk/reader/36484420.

28 Julian Gutierrez, Aniello Murano, Giuseppe Perelli, Sasha Rubin, Thomas Steeples, and Michael J. Wooldridge. Equilibria for games with combined qualitative and quantitative objectives. *Acta Informatica*, 58(6):585–610, 2021. doi:10.1007/s00236-020-00385-4.

29 Julian Gutierrez, Muhammad Najib, Giuseppe Perelli, and Michael J. Wooldridge. Automated temporal equilibrium analysis: Verification and synthesis of multi-player games. *Artificial Intelligence*, 287:103353, 2020. doi:10.1016/j.artint.2020.103353.

30 Julian Gutierrez, Muhammad Najib, Giuseppe Perelli, and Michael J. Wooldridge. On the complexity of rational verification. *Ann. Math. Artif. Intell.*, 91(4):409–430, 2023. doi:10.1007/s10472-022-09804-3.

31 Bernard Ralph Hodgson. Décidabilité par automate fini. *Annales des sciences mathématiques du Québec*, 7:39–57, 1983. URL: https://www.labmath.uqam.ca/~annales/volumes/07-1/PDF/039-057.pdf.

32 Orna Kupferman. Automata Theory and Model Checking. In Edmund M. Clarke, Thomas A. Henzinger, Helmut Veith, and Roderick Bloem, editors, *Handbook of Model Checking*, pages 107–151. Springer, 2018. doi:10.1007/978-3-319-10575-8_4.

33 Orna Kupferman, Giuseppe Perelli, and Moshe Y. Vardi. Synthesis with rational environments. *Annals of Mathematics and Artificial Intelligence*, 78(1):3–20, 2016. doi:10.1007/s10472-016-9508-8.

34 Stéphane Le Roux and Arno Pauly. Equilibria in multi-player multi-outcome infinite sequential games. *Information and Computation*, 276:104557, 2021. 5th International Workshop on Strategic Reasoning (SR 2017). doi:10.1016/j.ic.2020.104557.

35 Christof Löding and Christopher Spinrath. Decision Problems for Subclasses of Rational Relations over Finite and Infinite Words. *Discrete Mathematics & Theoretical Computer Science*, 21(3), 2019. doi:10.23638/dmtcs-21-3-4.

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

37 Martin J. Osborne. *An introduction to game theory*. Oxford Univ. Press, 2004.

38 Martin J. Osborne and Ariel Rubinstein. *A course in game theory*. The MIT Press, Cambridge, USA, 1994. electronic edition.

39 Sasha Rubin. Automatic structures. In Jean-Éric Pin, editor, *Handbook of Automata Theory*, pages 1031–1070. European Mathematical Society Publishing House, Zürich, Switzerland, 2021. doi:10.4171/automata-2/6.

40 Jacques Sakarovitch. *Elements of Automata Theory*. Cambridge University Press, 2009. URL: http://www.cambridge.org/uk/catalogue/catalogue.asp?isbn=9780521844253.

41 Wolfgang Thomas. Automata on Infinite Objects. In Jan Van Leeuwen, editor, *Formal Models and Semantics*, Handbook of Theoretical Computer Science, pages 133–191. Elsevier, Amsterdam, 1990. doi:10.1016/B978-0-444-88074-1.50009-3.

42 Michael Ummels. The Complexity of Nash Equilibria in Infinite Multiplayer Games. In Roberto 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.

43 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.