# Games with $\omega$-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 $\omega$-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 $\omega$-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 $\omega$-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, $\omega$-automatic relations, $\omega$-recognizable relations, constrained Nash equilibria existence problem

**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., [11, 12, 16, 26, 31, 34, 37]. 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* [40, 41], 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* [30] 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 $\omega$-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., [3]).

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 [3, 5, 10], and here we demonstrate how this idea can be adapted to fit the non-zero-sum game setting.

**Contributions.** Our contributions center on using *$\omega$-automatic relations* on infinite words [49], as introduced in [27], 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 [40] with $\omega$-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 $\cap$ 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 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.

[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., [23, 33])

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

Additionally, we analyze the algorithmic complexity of verifying whether an $\omega$-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 $\omega$-automatic preference relations are all *$\omega$-recognizable* (a strict subclass of $\omega$-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 [19, 47]. The $\omega$-automatic relations — also called synchronized $\omega$-rational relations — were first studied in [27]. Some decision problems about $\omega$-automatic and $\omega$-recognizable relations were solved in [39] and improved in [4]. The study of *automatic structures* has also led to results involving rational relations, notably within first-order logic (see, e.g., [6, 29, 35, 45]).

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., [8, 32, 50, 51]. 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 [14], the authors prove the existence of finite-memory NEs for all cost-semi-linear reward functions. In [9], 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 [9] are all $\omega$-automatic. In [24], 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 $\omega$-regular, the reward functions of [24] lead to $\omega$-automatic preference relations. The existence of NEs is guaranteed within a broad setting, both in [31] and [38], without relying on an automata-based approach, however with no complexity result about the constrained NE existence problem. In case of games with $\omega$-recognizable preference relations, our proof that NEs always exist relies on the technique developed in [31].

The results we obtain with games with $\omega$-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 [3] that the related preference relation is not $\omega$-automatic; and in the second case, the constrained NE existence problem is undecidable [13]. Note that the general concepts of $\omega$-automatic and $\omega$-recognizable relations have also been used to study imperfect information in games in [5, 10] and formal verification of quantitative systems in [3].

## 2 Preliminaries

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

**Automatic Relations.** Let $\Sigma$ be a fixed finite alphabet. We consider *binary relations* $R \subseteq \Sigma^\omega \times \Sigma^\omega$ on infinite words over $\Sigma$. The relation $R$ is *$\omega$-automatic* if it is accepted by a deterministic finite parity automaton over the alphabet $\Sigma \times \Sigma$, that is, $R$ is an $\omega$-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 *$\omega$-recognizable* if it is equal to $\cup_{i=1}^{\ell} X_i \times Y_i$ where $X_i, Y_i \subseteq \Sigma^\omega$ are $\omega$-regular languages over $\Sigma$ [39]. Any $\omega$-recognizable relation is $\omega$-automatic [47].

We suppose that the reader is familiar with the usual notion of deterministic parity automaton (DPA) used to accept $\omega$-automatic relations [39]. 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., [30] for general definitions, or [2, 36] 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 $\pi_k, \pi_{k+1}$. The set of all plays of an arena $A$ is denoted $\mathsf{Plays}_A \subseteq V^\omega$, and we write $\mathsf{Plays}$ when the context is clear. The *length* $|h|$ of a history $h$ is the number of its vertices. The empty history is denoted $\varepsilon$.

A *game* $\mathcal{G} = (A, (R_i)_{i\in\mathcal{P}})$ is an arena equipped with $\omega$-automatic relations $R_i$ over the alphabet $V$, one for each player $i$, called his *preference relation*. For any two plays $\pi, \pi'$, player $i$ *prefers* $\pi'$ to $\pi$ 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 $\omega$-automatic relation without any additional hypotheses. Such hypotheses will be discussed in Section 6.

Given a play $\pi$ and an index $k$, we write $\pi_{\geq k}$ the suffix $\pi_k\pi_{k+1}\dots$ of $\pi$. We denote the first vertex of $\pi$ 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 $\pi = \mu\nu\nu\cdots = \mu(\nu)^\omega$, where $\mu\nu$ is a history and $\nu$ a cycle, is called a *lasso*. The *length* of $\pi$ 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 $\sigma_i$ if $\pi_{k+1} = \sigma_i(\pi_0 \dots \pi_k)$ for all $k \in \mathbb{N}$ 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 $\sigma_i$ a strategy for player $i$, is called a *strategy profile*. The play $\pi$ starting from an initial vertex $v_0$ and consistent with each $\sigma_i$ is denoted by $\langle\sigma\rangle_{v_0}$ and called *outcome*.

A strategy $\sigma_i$ for player $i$ is *finite-memory* [30] 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 $\sigma_i$ such that $\sigma_i(hv) = \alpha_N(\widehat{\alpha}_U(m_0, h), v)$

[3] The negation is not allowed in the Boolean combination.

[4] To have a well-defined length for a lasso $\pi$, we assume that $\pi = \mu(\nu)^\omega$ with $\mu\nu$ of minimal length.

for all histories $hv \in V^*V_i$, where $\widehat{\alpha}_U$ extends $\alpha_U$ to histories as expected. A strategy $\sigma_i$ 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 $\omega$-regular objectives and of winning strategy [25, 30].

**Generality of the $\omega$-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 $\omega$-regular objective $\Omega_i \subseteq V^\omega$, such as a reachability or a Büchi objective [25, 30]. 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 $\Omega_i$ is seen as a function $\Omega_i : V^\omega \to \{0,1\}$. As $\Omega_i$ is $\omega$-regular, it follows that $\prec_i$ is $\omega$-automatic. For instance, given a target set $T \subseteq V$, the first DPA of Figure 1 accepts $\prec_i$ when $\Omega_i$ is a *reachability* objective $\{x = x_0x_1 \ldots \in V^\omega \mid \exists k, x_k \in T\}$; the second DPA accepts $\prec_i$ when $\Omega_i$ 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 $\omega$-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 $\bar{\Omega}_i(x) < \bar{\Omega}_i(y)$ [9]. 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 [9] are $\omega$-automatic.

Let us move on to classical quantitative objectives, such as quantitative reachability, limsup or discounted-sum objectives [25, 30]. 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 [3] that such a relation is $\omega$-automatic for a limsup objective and for a discounted-sum objective with an integer discount factor. They also proved that $\prec_i$ is not $\omega$-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 $\ell$ 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 $\omega$-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).

## 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 $\tau_i$ of player $i$, we have $\langle\sigma\rangle_{v_0} \not\prec_i \langle\tau_i, \sigma_{-i}\rangle_{v_0}$, where $\sigma_{-i}$ denotes $(\sigma_j)_{j\in\mathcal{P}\setminus\{i\}}$. So, NEs are strategy profiles where no single-player has an

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

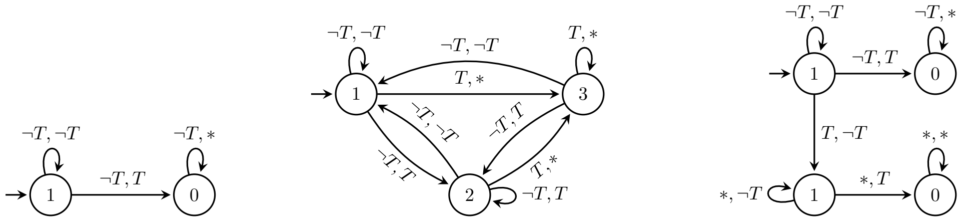

**Figure 1** DPAs accepting preference relations, corresponding respectively to reachability, Büchi, and max-reward-reachability objectives. The priorities are indicated inside each state, and an edge label $T$, $\neg T$, or $*$ means that there is an edge for each label $v \in T$, $v \in V \setminus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

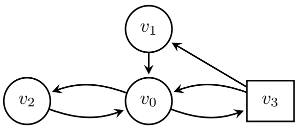

**Figure 2** An arena with round (resp. square) vertices owned by player 1 (resp. player 2).

incentive to unilaterally deviate from his strategy. When there exists a strategy $\tau_i$ such that $\langle\sigma\rangle_{v_0} \prec_i \langle\tau_i, \sigma_{-i}\rangle_{v_0}$, we say that $\tau_i$ (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 $\langle\sigma\rangle_{v_0} = (v_0v_3)^\omega$. Player 1 has no profitable deviation if player 2 sticks on his strategy $\sigma_2$: 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,
- $\sigma'_2$ is the memoryless strategy such that $\sigma'_2(v_3) = v_1$.

In that case, the NE outcome is $\langle\sigma'\rangle_{v_0} = v_0v_3v_1(v_0v_2)^\omega$. 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 $\Omega_1^1$ with $T_1$ and a Büchi objective $\Omega_1^2$ 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 $\Omega_1^2(x) < \Omega_1^2(y))$. If we consider the two previous strategy profiles, $\sigma'$ is still an NE, but $\sigma$ is no longer an NE as player 1 has a profitable deviation. Indeed, with the memoryless strategy $\tau_1$ 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 $\omega$-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

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

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 $\sigma_i$ is defined by a Mealy machine $\mathcal{M}_i$, whether $\sigma$ 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 $\pi$, whether $\pi$ 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 $\pi_i$ for each player $i$, whether there exists an NE from $v_0$ in $\mathcal{G}$ with an outcome $\rho$ 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 $\omega$-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 *$\omega$-recognizable* relations.

## 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 Appendix A.

**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 $\sigma_i$ 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

[8] A similar example is given in [38].

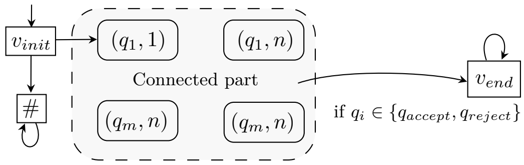

**Figure 3** The game used for PSPACE-hardness of Theorem 3.

there exists a strategy $\tau_i$ for some player $i$ such that $\langle \sigma \rangle_{v_0} \prec_i \langle \tau_i, \sigma_{-i} \rangle_{v_0}$ (in which case $\sigma$ 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 $\sigma_{-i}$ 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 $\delta$ is such that $\delta((v, (m_j)_{j \neq i}), v') = (v', (m'_j)_{j \neq i})$ for $\alpha^j_U(m_j, v) = m'_j$, and $\delta(s_0, v_0) = (v_0, (m^j_0)_{j \neq i})$. Note that $\delta$ 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 $\nu$ 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 [28], 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, $i \in \{1, \dots, n\}$, 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 $\pi$ starting at $v_0$, checking whether $\pi$ is an NE outcome amounts to finding a strategy profile $\sigma = (\sigma_i)_{i \in \mathcal{P}}$ with outcome $\pi$ such that for all $i \in \mathcal{P}$ and all strategies $\tau_i$, we have $\pi \not\prec_i \langle \tau_i, \sigma_{-i} \rangle_{v_0}$. In other words, given $\sigma$ a strategy profile partially defined such that $\pi = \langle \sigma \rangle_{v_0}$, our goal is to check whether, for all $i$, there exists $\sigma_{-i}$ that extends this partially defined profile such that for all $\tau_i$, $\pi \not\prec_i \langle \tau_i, \sigma_{-i} \rangle_{v_0}$. 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 $\pi$. Clearly, the size of $\mathcal{B}_i$ is polynomially bounded in the sizes of $\mathcal{A}_i$ and $\pi$. So, $L_i$ contains all the deviations that are not profitable for player $i$ compared to $\pi$. Now, it suffices to decide whether the coalition $-i$ has a strategy $\sigma_{-i}$ against player $i$ to ensure that every play consistent with $\sigma_{-i}$ 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 [30].

To obtain an algorithm in $\mathsf{NP}$, it remains to check whether $\pi$, 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 $\tau_B$ from $(v, q)$ for player $B$, a strategy $\sigma_{-i}$ for the coalition $-i$ such that for all $\tau_i$, $\pi \not\prec_i \langle \tau_i, \sigma_{-i} \rangle_{v_0}$. Similarly, to obtain an algorithm in $\mathsf{coNP}$, we check whether $\pi$ in $\mathcal{H}_i$ crosses at least one vertex $(v, q)$ that is winning for player $A$ and deduce a winning strategy for player $i$.

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 $\mathsf{parity}$-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 $\alpha : V \to \{0, \ldots, d\}$. 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

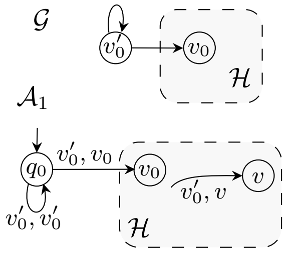

**Figure 4** The game $\mathcal{G}$ and the DPA $\mathcal{A}_1$ accepting $\prec_1$ for the reduction of Theorem 4.

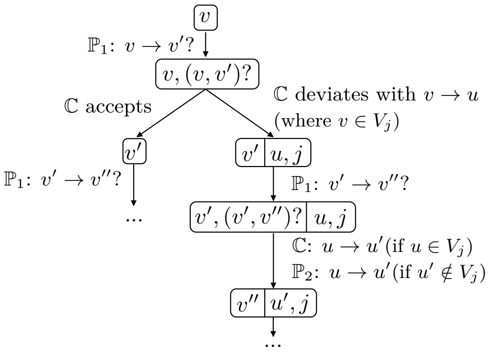
$v$
$\mathbb{P}_1$: $v \to v'$?
$v, (v, v')?$
$\mathbb{C}$ accepts
$\mathbb{C}$ deviates with $v \to u$
(where $v \in V_j$)
$v'$
$v'$ $u, j$
$\mathbb{P}_1$: $v' \to v''$?
$\mathbb{P}_1$: $v' \to v''$?
...
$v', (v', v'')?$ $u, j$
$\mathbb{C}$: $u \to u'$(if $u \in V_j$)
$\mathbb{P}_2$: $u \to u'$(if $u' \notin V_j$)
$v''$ $u', j$
...

**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 $\rho \not\prec_j \rho'$.

$\mathcal{H}$ starting at $v_0$ and satisfying the parity condition $\alpha$. 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 $\pi$, there exists a deviating strategy $\tau_1$ of player 1 such that $\pi \prec_1 \rho$ with $\rho = \langle \tau_1, \sigma_2 \rangle_{v_0'}$. Thus, by definition of $\prec_1$, $\rho$ is equal to $(v_0')^m \rho'$ with $\rho'$ a winning play in $\mathcal{H}$. Hence, transferred to $\mathcal{H}$, we get that for each strategy $\sigma_2'$ of player 2, there exists a strategy $\tau_1'$ 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 $\pi$. ◀

## 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 [17]. 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 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 $\rho$ while Challenger contests that it is an NE outcome: $\mathbb{P}_1$ has the task of building $\rho$ edge by edge, while $\mathbb{P}_2$ has the task of showing that the deviation $\rho'$ 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 $\rho$ cannot depend on one specific deviation and must be fixed, i.e., its construction cannot change according to the deviation $\rho'$ to artificially force $\rho \not\prec_i \rho'$. This also means that $\mathbb{P}_1$ has to build $\rho$ without knowing when $\mathbb{C}$ deviates: he has *partial observation* of the game, while $\mathbb{C}$ and $\mathbb{P}_2$ have

[9] Although the underlying game structure of [17] is reused, the player roles and correctness arguments differ entirely.

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 $\rho$, and an action of $\mathbb{C}$ is either to accept it or to deviate from $\rho$ 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 $\rho$, the construction of the deviation $\rho'$, and the component $j$ to identify the player who deviated: $\mathbb{P}_1$ continues to propose an extension $(v, v')$ for $\rho$ 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 $\rho'$. When the game stays in the first part, the objective of $\mathbb{P}_1$ is to produce an NE outcome $\rho$, 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 $\rho'$ 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 $\rho$ with the deviation $\rho'$.

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

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

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 $v, u \in V$, $j \in \mathcal{P} \cup \{\bot\}$, $q_i \in Q_i$, and $(v, v') \in E$. Coming back to the intuition given above, $v$ is the current vertex of $\rho$, $j$ is the deviating player (or $\bot$ if $\mathbb{C}$ did not deviate yet), $u$ is the current vertex of $\rho'$ (if it exists, otherwise $u = v$), $q_i$ is the current state of $\mathcal{A}_i$ while comparing $\rho$ and $\rho'$.
  Given that we are looking for an NE in $\mathcal{G}$ from some initial vertex $v_0$, we consider the initial vertex $s_0 = (v_0, \bot, v_0, (q_i^0)_i)$ 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 $j = \bot$ and $v = u$, or $j \neq \bot$ 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 $j \neq \bot$ and $u \in V \backslash V_j$.
- 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 $\rho$) and $A_{\mathbb{C}} = A_{\mathbb{P}_2} = V$ ($\mathbb{C}$ and $\mathbb{P}_2$ choose the next vertex $u$ of $\rho'$ in case $\mathbb{C}$ deviates, otherwise $\mathbb{C}$ accepts the vertex $v'$ of the edge $(v, v')$ proposed by $\mathbb{P}_1$.)
- 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.

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

  - 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 $s_0$, $\pi$ 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 $\pi$, 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 $\pi$, either the second component always remains equal to $\perp$ or ultimately becomes equal to some $j \in \mathcal{P}$. We use notation $\mathsf{dev}(\pi)$ to denote this value $\perp$ 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.,
  - $W_{acc} = \{\pi \in \mathsf{Plays}(\mathbb{P}_1\mathbb{C}\mathbb{P}_2(\mathcal{G})) \mid \mathsf{dev}(\pi) = \perp\}$,
  - $W_{dev} = \{\pi \in \mathsf{Plays}(\mathbb{P}_1\mathbb{C}\mathbb{P}_2(\mathcal{G})) \mid \exists j \in \mathcal{P}, \mathsf{dev}(\pi) = j \text{ and } \mathsf{proj}_{V,1}(\pi) \not\prec_j \mathsf{proj}_{V,2}(\pi)\}$.

  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 $\langle \tau_{\mathbb{P}_1}, \tau_{\mathbb{C}}, \tau_{\mathbb{P}_2} \rangle_{s_0} \in W_{\mathbb{P}_1\mathbb{P}_2}$.*

Theorem 7 is the key tool to solve the NE existence problem. It is proved in detail in Appendix B. We give hereafter a sketch of proof for the membership result of Theorem 5, which follows the approach proposed in [17]. 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 Appendix C.

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

**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 [17], 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 [17]. 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 [36]. Steps *(ii)* and *(iii)* are detailed in Appendix C, 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 Appendix D. 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 $\pi_i$ 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 $\rho$ 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.

## 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

[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 $\pi$ such that there exists $j \in J$ with $\mathsf{Inf}(\pi) \cap E_j = \varnothing$ and $\mathsf{Inf}(\pi) \cap F_j \neq \varnothing$.

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 Appendix E.

▶ **Proposition 9.** *The problem of deciding whether an $\omega$-automatic relation $R$ is reflexive (resp. irreflexive, transitive, $\neg$-transitive, total) is* $\mathsf{NL}$*-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 $\omega$-automatic, $\precsim$ is also $\omega$-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 $\omega$-automatic, then $\prec$ and $\sim$ are both $\omega$-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 $\mathsf{NP}$ 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 $\mathsf{NP}$ [22]. All lower bounds remain valid by carefully modifying the preference relations used in the reductions into preorders, see Appendix F.

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

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

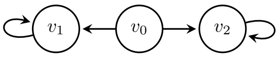

**Figure 6** An arena with one player.

**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 $\sigma$ such that for all players $i$ and all strategies $\tau_i$ 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 $\tau_i$ that $\langle\tau_i, \sigma_{-i}\rangle_{v_0} \precsim_i \langle\sigma\rangle_{v_0}$ [9]. 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 $\omega$-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 *$\omega$-recognizable* and *preorders*. We recall that $\precsim_i$ is $\omega$-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 $\omega$-regular languages over $\Sigma$. Any $\omega$-recognizable relation is $\omega$-automatic (see [47]), and deciding whether an $\omega$-automatic relation accepted by a DPA is $\omega$-recognizable is NL-complete [4]. 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 $\omega$-recognizable preorders, as highlighted in the next proposition, easily derived from [39] (its proof is given in Appendix G). This motivates the interest of games with $\omega$-recognizable preference relations.

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

Thanks to this result, we can partition $\Sigma^\omega$ 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 $\omega$-recognizable preorders are numerous: those deriving from any Boolean combination of $\omega$-regular objectives or any multidimensional objective where each dimension is defined using an $\omega$-regular objective. In the subclass of games with $\omega$-recognizable preorders, the main difference is the *existence of NEs.*

▶ **Theorem 11.** *When the preference relations of a game are all $\omega$-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 [38, 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 Appendix H, inspired by the work of [31] and [14], where the existence of NEs is studied through the concept of value and optimal strategy (see below and in Appendix H).

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 Appendix I).

▶ **Proposition 12.** *Any $\omega$-recognizable preorder $\precsim$ can be embedded into an $\omega$-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 Appendix J). 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* $\sigma_i^v$ for player $i$ and $\sigma_{-i}^v$ for the coalition $-i$ such that $\sigma_i^v$ (resp. $\sigma_{-i}^v$) ensures consistent plays $\pi$ starting at $v$ such that $val_i(v) \precsim_i [\pi]_i$ (resp. $[\pi]_i \precsim_i val_i(v)$) (see Appendix H). Such an NE characterization is well-known for games with classical objectives (see, e.g., the survey [15]).

▶ **Theorem 13.** *Let $\mathcal{G}$ be a game such that each preference relation $\precsim_i$ is an $\omega$-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 $\rho_n$ of $\rho$, 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 Appendix J.

## 8 Conclusion

In this work, we have introduced a general framework for defining players' preferences via $\omega$-automatic preference relations instead of fixed reward functions. It subsumes several classical settings, including the Boolean setting with $\omega$-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 [17]. 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 [42]), or the rational synthesis problem as studied in [37]. 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 Sanjeev Arora and Boaz Barak. *Computational Complexity - A Modern Approach*. Cambridge University Press, 2009. URL: `https://theory.cs.princeton.edu/complexity/`.

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

3 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.

4 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`.

5 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`.

6 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`.

7 Udi Boker, Orna Kupferman, and Avital Steinitz. Parityizing Rabin and Streett. In Kamal Lodaya and Meena Mahajan, editors, *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2010, December 15-18, 2010, Chennai, India*, volume 8 of *LIPIcs*, pages 412–423. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2010. `doi:10.4230/lipics.fsttcs.2010.412`.

8 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`.

9 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`.

10 Laura Bozzelli, Bastien Maubert, and Sophie Pinchinat. Uniform strategies, rational relations and jumping automata. *Information and Computation*, 242, 2015. URL: `https://www.sciencedirect.com/science/article/pii/S0890540115000279`, `doi:10.1016/j.ic.2015.03.012`.

11 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. URL: `https://doi.org/10.1007/978-3-319-30000-9%5F1`, `doi:10.1007/978-3-319-30000-9\_1`.

12 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`.

13 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`.

14 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. URL: `https://doi.org/10.1007/978-3-642-35722-0%5F5`, `doi:10.1007/978-3-642-35722-0\_5`.

15 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. URL: https://doi.org/10.1007/978-3-319-62809-7%5F1, doi:10.1007/978-3-319-62809-7_1.

16 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.

17 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.

18 Cristian S. Calude, Sanjay Jain, Bakhadyr Khoussainov, Wei Li, and Frank Stephan. Deciding Parity Games in Quasi-polynomial Time. *SIAM Journal on Computing*, 51(2):STOC17–152–STOC17–188, 2022. doi:10.1137/17M1145288.

19 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. URL: http://www.numdam.org/articles/10.1051/ita:2006005, doi:10.1051/ita:2006005.

20 Krishnendu Chatterjee and Laurent Doyen. The Complexity of Partial-Observation Parity Games. In Christian G. Fermüller and Andrei Voronkov, editors, *Logic for Programming, Artificial Intelligence, and Reasoning - 17th International Conference, LPAR-17, Yogyakarta, Indonesia, October 10-15, 2010. Proceedings*, volume 6397 of *Lecture Notes in Computer Science*, pages 1–14. Springer, 2010. URL: https://doi.org/10.1007/978-3-642-16242-8%5F1, doi:10.1007/978-3-642-16242-8_1.

21 Krishnendu Chatterjee and Laurent Doyen. Games with a Weak Adversary. In Javier Esparza, Pierre Fraigniaud, Thore Husfeldt, and Elias Koutsoupias, editors, *Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part II*, volume 8573 of *Lecture Notes in Computer Science*, pages 110–121. Springer, 2014. URL: https://doi.org/10.1007/978-3-662-43951-7%5F10, doi:10.1007/978-3-662-43951-7_10.

22 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. URL: https://doi.org/10.1007/978-3-540-71389-0%5F12, doi:10.1007/978-3-540-71389-0_12.

23 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.

24 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.

25 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.

26 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.

**27** Christiane Frougny and Jacques Sakarovitch. Synchronized rational relations of finite and infinite words. *Theoretical Computer Science*, 108(1):45–82, 1993. URL: `https://www.sciencedirect.com/science/article/pii/030439759390230Q`, `doi:10.1016/0304-3975(93)90230-Q`.

**28** 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`.

**29** 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, page 21–34, New York, NY, USA, 2020. Association for Computing Machinery. `doi:10.1145/3373718.3394734`.

**30** 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`.

**31** 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`.

**32** 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`.

**33** 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`.

**34** 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`.

**35** 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`.

**36** 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. URL: `https://doi.org/10.1007/978-3-319-10575-8%5F4`, `doi:10.1007/978-3-319-10575-8\_4`.

**37** 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`.

**38** 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`.

**39** 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`.

**40** 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`.

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

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

**43** Jean-François Raskin, Krishnendu Chatterjee, Laurent Doyen, and Thomas A. Henzinger. Algorithms for Omega-Regular Games with Imperfect Information. *Logical Methods in Computer Science*, 3(3), 2007. `doi:10.2168/LMCS-3(3:4)2007`.

**44** John H. Reif. The Complexity of Two-Player Games of Incomplete Information. *Journal of Computer and System Sciences*, 29(2):274–301, 1984. `doi:10.1016/0022-0000(84)90034-5`.

45 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.

46 Shmuel Safra. *Complexity of Automata on Infinite Objects*. PhD thesis, Weizmann Institute of Science, Rehovot, Israel, March 1989.

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

48 Michael Sipser. *Introduction to the Theory of Computation*. Course Technology, Boston, MA, third edition, 2013.

49 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.

50 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. URL: http://www.logic.rwth-aachen.de/~ummels/fossacs08.pdf, doi:10.1007/978-3-540-78499-9_3.

51 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. URL: https://doi.org/10.1007/978-3-642-23217-6%5F32, doi:10.1007/978-3-642-23217-6_32.

## A Hardness of Theorem 3

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

**Proof of Theorem 3, hardness.** We use a reduction from the membership problem for linear bounded deterministic Turing machines (LBTMs), also called Linear Bounded Automata [48], known to be PSPACE-complete [28], 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 cells containing the input word $w$. The membership problem asks whether $w$ is accepted by $T$.

Suppose that $T$ is defined on a set of states $Q = \{q_1, \dots, q_m\}$ with $q_1$ its initial state and containing $q_{accept}, q_{reject}$, the word $w$ is equal to $w_1 w_2 \dots w_n$, and the alphabet is $\Sigma = \{a_1, \dots, a_s\}$. We construct a $(n+1)$-player game $\mathcal{G} = (V, E, (V_i)_{1 \le i \le n+1}, (\prec_i)_{1 \le i \le n+1})$ as illustrated in Figure 3, and $n+1$ Mealy machines $(\mathcal{M}_i)_{1 \le i \le n+1}$ in the following way. Intuitively, each input tape cell is associated with a player and will be simulated by a Mealy machine for this player, encoding actions based on the current configuration of the LBTM. We begin with the description of the game:

- $V = (Q \times \{1, \dots, n\}) \cup \{v_{init}, \#, v_{end}\}$, with three fresh vertices $v_{init}$, $\#$, and $v_{end}$,
- $v_{init}$ is the initial vertex and has two successors: $\#$ and $(q_1, 1)$,
- $\#$ and $v_{end}$ are sink vertices with a self-loop,
- $V_{n+1} = \{v_{init}, \#, v_{end}\}$ and for each $i \in \{1, \dots, n\}$, $V_i = Q \times \{i\}$,
- $(q, i) \to (q', j)$ for all $q, i, q', j$ such that $q \notin \{q_{accept}, q_{reject}\}$, otherwise $(q, i) \to v_{end}$,
- for each $i \in \{1, \dots, n\}$, the preference relation $\prec_i$ is empty, therefore accepted by a one-state DPA,

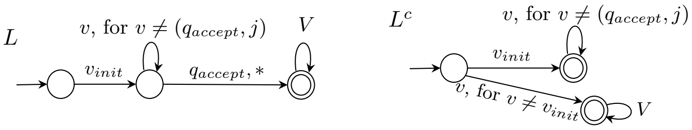
$L$
$v$, for $v \neq (q_{accept}, j)$
$V$
$v_{init}$
$q_{accept}, *$
$L^c$
$v$, for $v \neq (q_{accept}, j)$
$v_{init}$
$v$, for $v \neq v_{init}$
$V$

**Figure 7** The DBAs accepting the languages $L$ and $L^c$ used to define the preference relation $\prec_{n+1}$ in the game of Figure 3.

- the preference relation $\prec_{n+1}$ is defined such that $v_{init}\#^\omega \prec_{n+1} v_{init}x$ if and only if $x$ contains $\{q_{accept}\} \times \{1, \dots, n\}$. More precisely, we define $\prec_{n+1} = L^c \times L$ where $L = \{v_{init}x \in V^\omega \mid x \text{ visits } \{q_{accept}\} \times \{1, \dots, n\}\}$ and $L^c = V^\omega \setminus L$.

Finally, let us show that $\prec_{n+1}$ is accepted by a DBA of constant size. Both languages $L$ and $L^c$ are accepted by a DBA as illustrated in Figure 7. Thus, the relation $L^c \times L$ is accepted by a generalized DBAs with a conjunction of two Büchi conditions of constant size, thus by a DBA which is in particular a DPA.

Let us now describe the Mealy machines. The Mealy machine of player $n+1$ defines the memoryless strategy $\sigma_{n+1}$ such that $\sigma_{n+1}(v_{init}) = \#$. For player $i \in \{1, \dots, n\}$, the Mealy machine $\mathcal{M}_i$ simulates the LBTM $T$ when the tape head scans the $i$-th tape cell:

- the set of memory states is $M = \Sigma \cup \{a_{init}, a_{end}\}$, with fresh symbols $a_{init}, a_{end} \notin \Sigma$, where $a_{init}$ is the initial memory state,
- the udpate function $\alpha_U : M \times V \to M$ indicates which symbol is written by $T$ in the $i$-th cell when the tape head scans this cell; it is defined as follows:
  - from the state $a \neq a_{init}, a_{end}$ and the vertex $(q, j) \in V$, stay on state $a$ if $j \neq i$, or go to $a'$ if $j = i$ and $(q, a) \to (q', a', Move)$ is a transition of $T$, or go to $a_{end}$ if $q \in \{q_{accept}, q_{reject}\}$.
  - from the state $a_{end}$, the only available vertex is $v_{end}$; thus stay on $a_{end}$.
  - from the state $a_{init}$, the only available vertex is $v_{init}$; thus go to state $a$ such that $w_i = a$.
- the next-move function $\alpha_N : M \times V_i \to V$ (recall that $V_i = Q \times \{i\}$) indicates the move ($R$ or $L$) of $T$ when the tape head scans the $i$-th cell; it is defined as follows. From the state $a \neq a_{init}, a_{end}$ and vertex $(q, i)$, output $(q', j)$ such that $(q, a) \to (q', a', Move)$ is a transition of $T$ and $j = \min(n, i+1)$ if $Move = R$, $j = \max(0, i-1)$ if $Move = L$.

By the construction given above, we know that $(\mathcal{M}_i)_i$ produces the outcome $v_{init}\#^\omega$. As player $n+1$ controls $v_{init}$, no deviation of player $i \in \{1, \dots, n\}$ modifies this outcome, i.e., these deviations are not profitable. However, player $n+1$ has exactly one possible deviation, to go from $v_{init}$ to $(q_1, 1)$, thus leading to some play $v_{init}\pi$. This deviation is profitable if $v_{init}\#^\omega \prec_{n+1} v_{init}\pi$, and we know by construction that $v_{init}\#^\omega \prec_{n+1} v_{init}\pi$ if and only if $\pi$ visits the state $q_{accept}$, i.e., $T$ accepts $w$ (as $\pi$ represents an execution of the LBTM $T$ thanks to the definition of $\mathcal{M}_1, \dots, \mathcal{M}_n$). This shows the correctness of the reduction. Moreover, this is a polynomial reduction. Therefore, the NE checking problem is PSPACE-hard. ◀

## B Correctness of the $\mathbb{P}_1\mathbb{C}\mathbb{P}_2$ Game

▶ **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 $\langle \tau_{\mathbb{P}_1}, \tau_{\mathbb{C}}, \tau_{\mathbb{P}_2} \rangle_{s_0} \in W_{\mathbb{P}_1\mathbb{P}_2}$.*

Before proving this result, we study in more details the correspondence between strategies in $\mathcal{G}$ and strategies in $\mathbb{P}_1\mathbb{C}\mathbb{P}_2(\mathcal{G})$. We first note that if a strategy $\tau_{\mathbb{P}_1}$ of $\mathbb{P}_1$ is observation-based, we get that for all plays $\pi, \pi'$ starting at $s_0$ and consistent with $\tau_{\mathbb{P}_1}$,

$$\mathsf{proj}_{V,1}(\pi) = \mathsf{proj}_{V,1}(\pi') \text{ and } \mathsf{proj}_E(\pi) = \mathsf{proj}_E(\pi'). \tag{1}$$

Given a play $\rho$ of $\mathcal{G}$ starting at $v_0$, we call *simulation of* $\rho$ any strategy $\tau_{\mathbb{P}_1}$ of $\mathbb{P}_1$ such that $\rho = \mathsf{proj}_{V,1}(\pi)$ for all plays $\pi$ starting at $s_0$ and consistent with $\tau_{\mathbb{P}_1}$.

▶ **Lemma 14.** *Let $\rho$ be a play of $\mathcal{G}$ starting at $v_0$. There always exists a simulation $\tau_{\mathbb{P}_1}$ of $\rho$ that is observation-based.*

**Proof.** Consider a play $\rho = \rho_0\rho_1\dots$ of $\mathcal{G}$ starting at $v_0$. The required strategy $\tau_{\mathbb{P}_1}$ is constructed as follows. For all histories $hs$ such that $s \in S_{\mathbb{P}_1}$, we define $\tau_{\mathbb{P}_1}(hs) = (\rho_k, \rho_{k+1})$ if $\mathsf{proj}_{V,1}(hs) = \rho_0 \dots \rho_k$ is a prefix of $\rho$. Otherwise, we define $\tau_{\mathbb{P}_1}(hs)$ as an arbitrary action $e \in E$ such that $\tau_{\mathbb{P}_1}(h's') = e$ for all $h's'$ with $Obs(h's') = Obs(hs)$. By construction, $\tau_{\mathbb{P}_1}$ is observation-based because whenever $Obs(h's') = Obs(hs)$, then $\mathsf{proj}_{V,1}(h's') = \mathsf{proj}_{V,1}(hs)$. Moreover, for any play $\pi$ starting at $s_0$ and consistent with $\tau_{\mathbb{P}_1}$, we have $\rho = \mathsf{proj}_{V,1}(\pi)$. ◀

Let $\tau_{\mathbb{P}_1}$ be an observation-based strategy of $\mathbb{P}_1$ and $\rho$ be the play of $\mathcal{G}$ of which it is the simulation.

- We say that a strategy $\tau_{\mathbb{C}}$ of $\mathbb{C}$ is $\tau_{\mathbb{P}_1}$-*accepting* if for all histories $hs$ starting at $s_0$ and consistent with $\tau_{\mathbb{P}_1}$, if $\mathsf{proj}_E(s) = (v, v')$, then $\tau_{\mathbb{C}}(hs) = v'$. In this case, the plays $\pi$ consistent with $\tau_{\mathbb{P}_1}$ and $\tau_{\mathbb{C}}$ all satisfy $\mathsf{dev}(\pi) = \bot$.
- Let $hs$ be a history starting at $s_0$ and consistent with $\tau_{\mathbb{P}_1}$ such that $\mathsf{dev}(h) = \bot$ and $\mathsf{dev}(hs) = j$. Then, all plays $\pi$ consistent with $\tau_{\mathbb{P}_1}$ having $hs$ as prefix, satisfy $\mathsf{dev}(\pi) = j$ in addition to (1). Thus, only the third component of the vertices of those $\pi$ can vary.[14] Therefore, when $\tau_{\mathbb{P}_1}$ is fixed:
  - Any strategy $\tau_{\mathbb{C}}$ is equivalent to a strategy $\sigma_j$ of player $j$ such that $\tau_{\mathbb{C}}(h's') = \sigma_j(\mathsf{proj}_{V,2}(h's'))$ for all histories $h's'$ ending in $s' \in S_{\mathbb{C}}$ and having $hs$ as prefix. It is called a $\sigma_j$*-deviation from* $\tau_{\mathbb{P}_1}$.
  - Any strategy $\tau_{\mathbb{P}_2}$ is equivalent to a strategy profile $\sigma_{-j} = (\sigma_i)_{i \neq j}$ such that $\tau_{\mathbb{P}_2}(h's') = \sigma_i(\mathsf{proj}_{V,2}(h's'))$ for all histories $h's'$ ending in $s' \in S_{\mathbb{P}_2}$ with $\mathsf{proj}_{V,1}(s) \in V_i$ and having $hs$ as prefix. It is called a $\sigma_{-j}$*-punishment* strategy.

**Proof of Theorem 7.** Let us suppose that there exists an NE $\sigma = (\sigma_i)_{i\in\mathcal{P}}$ from $v_0$ in $\mathcal{G}$ and let $\rho = \langle\sigma\rangle_{v_0}$ be its outcome. By Lemma 14, there exists an observation-based strategy $\tau_{\mathbb{P}_1}$ of $\mathbb{P}_1$ such that for any play $\pi$ starting at $s_0$ and consistent with $\tau_{\mathbb{P}_1}$, we have $\mathsf{proj}_{V,1}(\pi) = \rho$. Let $\tau_{\mathbb{C}}$ be a strategy of $\mathbb{C}$.

- If $\tau_{\mathbb{C}}$ is $\tau_{\mathbb{P}_1}$-accepting, then we get $\mathsf{dev}(\pi) = \bot$ for all plays $\pi$ starting at $s_0$ and consistent with both $\tau_{\mathbb{P}_1}$ and $\tau_{\mathbb{C}}$, and any strategy $\tau_{\mathbb{P}_2}$, i.e., $\langle\tau_{\mathbb{P}_1}, \tau_{\mathbb{C}}, \tau_{\mathbb{P}_2}\rangle_{s_0} \in W_{acc}$.
- Otherwise, $\tau_{\mathbb{C}}$ is a $\sigma'_j$-deviation from $\tau_{\mathbb{P}_1}$, for a strategy $\sigma'_j$ of some player $j$. Consequently, we define $\tau_{\mathbb{P}_2}$ from the strategy profile $\sigma_{-j} = (\sigma_i)_{i\neq j}$ such that $\tau_{\mathbb{P}_2}$ is a $\sigma_{-j}$-punishment strategy. Therefore, for $\pi' = \langle\tau_{\mathbb{P}_1}, \tau_{\mathbb{C}}, \tau_{\mathbb{P}_2}\rangle_{s_0}$, we get $\mathsf{proj}_{V,2}(\pi') = \langle\sigma'_j, \sigma_{-j}\rangle_{v_0}$. As $\sigma$ is an NE by hypothesis, we have $\mathsf{proj}_{V,1}(\pi') = \rho \not\prec_j \mathsf{proj}_{V,2}(\pi')$, meaning that $\pi' \in W_{dev}$.

We now suppose that 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 $\langle\tau_{\mathbb{P}_1}, \tau_{\mathbb{C}}, \tau_{\mathbb{P}_2}\rangle_{s_0} \in W_{\mathbb{P}_1\mathbb{P}_2}$.

[14] The fourth component equal to $(q_i)_i$ derive from the first and third components.

Let us explain how to deduce an NE $\sigma = (\sigma_i)_i$ from $v_0$ in $\mathcal{G}$. We choose a strategy $\tau_{\mathbb{C}}$ that is $\tau_{\mathbb{P}_1}$-accepting, which gives by hypothesis a strategy $\tau_{\mathbb{P}_2}$ of $\mathbb{P}_2$. Observe that the resulting outcome $\pi = \langle \tau_{\mathbb{P}_1}, \tau_{\mathbb{C}}, \tau_{\mathbb{P}_2} \rangle_{s_0}$ is such that $\mathsf{dev}(\pi) = \perp$, so $\pi \in W_{acc}$. Let us define $\rho = \mathsf{proj}_{V,1}(\pi)$, it will be the outcome of the NE $\sigma$ we want to construct.

We now partially define the strategy profile $\sigma$ in a way to produce the outcome $\rho$. It remains to define $\sigma_i(hv)$ for all $i \in \mathcal{P}$ and all histories $hv$ which are not prefix of $\rho$.

Consider a strategy $\sigma'_j$ for player $j$ such that there exists a history $hv$, with $v \in V_j$, prefix of $\rho$, but where $hv \cdot \sigma'_j(hv)$ is not. Then, we can consider $\tau_{\mathbb{C}}$ that is a $\sigma'_j$-deviation from $\tau_{\mathbb{P}_1}$. By hypothesis, there exists a strategy $\tau_{\mathbb{P}_2}$ such that $\pi' = \langle \tau_{\mathbb{P}_1}, \tau_{\mathbb{C}}, \tau_{\mathbb{P}_2} \rangle_{s_0} \in W_{\mathbb{P}_1,\mathbb{P}_2}$. Since $\mathsf{dev}(\pi') = j$, it must be the case that $\pi' \in W_{dev}$, i.e., $\mathsf{proj}_{V,1}(\pi') \not\prec_j \mathsf{proj}_{V,2}(\pi')$. Hence, we can complete the definition of $\sigma$ outside $\rho$ by $\sigma_{-j}$ thanks to the strategy $\tau_{\mathbb{P}_2}$ seen as a $\sigma_{-j}$-punishment strategy. It follows that $\mathsf{proj}_{V,2}(\pi') = \langle \sigma'_j, \sigma_{-j} \rangle_{v_0}$. Therefore, by (1), $\rho = \mathsf{proj}_{V,1}(\pi) = \mathsf{proj}_{V,1}(\pi') \not\prec_j \mathsf{proj}_{V,2}(\pi') = \langle \sigma'_j, \sigma_{-j} \rangle_{v_0}$, showing that $\langle \sigma'_j, \sigma_{-j} \rangle_{v_0}$ is not a profitable deviation. We conclude that the constructed $\sigma$ is an NE. ◀

## C NE Existence Problem

▶ **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).*

**Proof of Theorem 5, membership.** By Theorem 7, deciding whether in $\mathcal{G}$, there exists an NE from $v_0$ reduces to deciding whether 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 $\langle \tau_{\mathbb{P}_1}, \tau_{\mathbb{C}}, \tau_{\mathbb{P}_2} \rangle_{s_0} \in W_{\mathbb{P}_1\mathbb{P}_2}$. In [17], the authors solve the problem they study by solving a similar three-player game with imperfect information. They proceed with the following three steps.

1. The winning condition is translated into a Rabin condition on the arena of the $\mathbb{P}_1\mathbb{C}\mathbb{P}_2$ game.
2. Then, the three-player game is transformed into a two-player zero-sum Rabin game with imperfect information.
3. Finally, classical techniques to remove imperfect information are used to obtain a two-player zero-sum parity game with perfect information.

For our $\mathbb{P}_1\mathbb{C}\mathbb{P}_2$ game, we already explain how to translate the winning condition $W_{\mathbb{P}_1\mathbb{P}_2}$ into a Rabin condition (first step). With the notations of Section 5, this game has

- a number of states in $\mathcal{O}(|V|^4 \cdot |\mathcal{P}| \cdot \Pi_{i \in \mathcal{P}} |\mathcal{A}_i|)$,
- a number of actions in $\mathcal{O}(|V|^2)$
- a number of Rabin pairs in $\mathcal{O}(\Sigma_{i \in \mathcal{P}} d_i)$

Let us explain the second step. To fit the context of [17], we need to slightly modify our $\mathbb{P}_1\mathbb{C}\mathbb{P}_2$ game in three ways:

- We add dummy vertices such that the three players play in a turn-based way, i.e., according to the turn sequence $(\mathbb{P}_1\mathbb{C}\mathbb{P}_2)^\omega$.
- Plays and histories include actions, i.e., they are sequences alternating vertices and actions, such that the histories always end with a vertex.
- Due to the modification of plays and histories, the observation function $Obs$ is extended to actions, i.e., $Obs(a) = a$ is $a \in A_{\mathbb{P}_1}$ and $Obs(a) = \#$ is $a \in A_{\mathbb{C}} \cup A_{\mathbb{P}_2}$ ($\mathbb{P}_1$ only observes his own actions, the other actions are not visible[15]).

[15] The situation is a little different in [17] as $\mathbb{P}_1$ can also observe the actions of $\mathbb{P}_2$.

With these modifications, we can translate our $\mathbb{P}_1\mathbb{C}\mathbb{P}_2$ game into a two-player zero-sum Rabin game with imperfect information exactly as in [17] (this is derived from a transformation introduced in [21], specifically adapted to the $\mathbb{P}_1\mathbb{C}\mathbb{P}_2$ game context). The main idea is to merge the two Provers into a new single Prover $\mathbb{P}$. Imperfect information is used to ensure that this merging does not grant too much knowledge to $\mathbb{P}$. In order to let the new Prover have as much actions available as $\mathbb{P}_2$ and stay observation-based, his action set includes, in addition to the actions of $A_{\mathbb{P}_1}$, all functions from $S_{\mathbb{P}_2}$ to $A_{\mathbb{P}_2}$. The set of vertices and the Rabin objective remain the same as in the $\mathbb{P}_1\mathbb{C}\mathbb{P}_2$ game. This yields a new game, called $\mathbb{P}\mathbb{C}$ game,

- with the same number of vertices and Rabin pairs,
- but with an exponential blowup in the number of actions.

Details about the new game can be found in [17]. In a way to guarantee the equivalence between both $\mathbb{P}_1\mathbb{C}\mathbb{P}_2$ and $\mathbb{P}\mathbb{C}$ games, some technical properties must be satisfied by the observation function $\mathit{Obs}$ of the $\mathbb{P}_1\mathbb{C}\mathbb{P}_2$ game:

- The function $\mathit{Obs}$ must be strongly player-stable and thus player-stable (see Corollary 4.1.4 and Lemma 4.1.5 of [17]). This means that for any two histories $h = s_0a_0s_1a_1 \dots a_{k-1}s_k = h_1s_k$ and $h' = s'_0a'_0s'_1a'_1 \dots a'_{k-1}s'_k = h'_1s'_k$ such that $s_0 = s'_0$ is the initial vertex of the $\mathbb{P}_1\mathbb{C}\mathbb{P}_2$ game and $\mathit{Obs}(h_1) = \mathit{Obs}(h'_1)$, it must follow that each pair of vertices $s_\ell, s'_\ell$, $1 \leq \ell \leq k$, are owned by the same player and $\mathit{Obs}(s_k) = \mathit{Obs}(s'_k)$. It is easy to verify that $s_\ell, s'_\ell$ are owned by the same player as the players play in a turn-based way. Let us explain why $\mathit{Obs}(s_k) = \mathit{Obs}(s_k)$. If $s_{k-1}$ is owned by $\mathbb{P}_1$, then $\mathit{Obs}(s_{k-1}) = \mathit{Obs}(s'_{k-1}) = v$ for some $v$ and $a_{k-1} = \mathit{Obs}(a_{k-1}) = \mathit{Obs}(a'_{k-1}) = a'_{k-1} = (v, v')$ for some $(v, v')$. Hence, $\mathit{Obs}(s_k) = \mathit{Obs}(s'_k) = (v, v')$. If $s_{k-1}$ is owned by $\mathbb{C}$ or $\mathbb{P}_2$, then $\mathit{Obs}(s_{k-1}) = \mathit{Obs}(s'_{k-1}) = (v, v')$ for some $(v, v')$ and thus $\mathit{Obs}(s_k) = \mathit{Obs}(s'_k) = v'$.
- The function $\mathit{Obs}$ must be action-stable (see Lemma 3.4.3 of [17]). This means that given $\Delta(s_1, a) = s_2$ and $\Delta(s'_1, a') = s'_2$ such that $\mathit{Obs}(s_1) = \mathit{Obs}(s'_1)$, then
  - if $a = a'$, then $\mathit{Obs}(s_2) = \mathit{Obs}(s'_2)$;
  - if $a, a'$ are visible (i.e., are actions of $\mathbb{P}_1$) and $\mathit{Obs}(s_2) = \mathit{Obs}(s'_2)$, then $a = a'$.

  It is easy to verify this property when $\mathit{Obs}(s_1) = \mathit{Obs}(s'_1) = v$ for some $v$, as $s_1, s'_1$ are then owned by $\mathbb{P}_1$ and his actions $a, a'$ are of the form $(v, v')$ for some $v'$. This is also easy to verify when $\mathit{Obs}(s_1) = \mathit{Obs}(s'_1) = (v, v')$ for some $(v, v')$ as $s_1, s'_1$ are then owned by $\mathbb{C}$ or $\mathbb{P}_2$ whose actions $a, a'$ are not visible.

With these technical properties and arguments similar to those of [17], one can carefully verify that our $\mathbb{P}_1\mathbb{C}\mathbb{P}_2$ game can be transformed into an equivalent two-player zero-sum Rabin game with imperfect information exactly as in [17].

In the third step, the latter $\mathbb{P}\mathbb{C}$ game is transformed into an equivalent two-player zero-sum parity game with perfect information. The translation is exactly the same as in [17]. The idea to get rid of the imperfect information is to apply standard game-theoretic techniques [20, 43, 44], by (1) making the Rabin condition visible, that is, such that any two similarly observed plays agree on the winning condition, and by (2) applying the subset construction to recall the set of possible visited vertices, and letting them be observed. Done carefully, this leads to a parity game with

- a number of vertices exponential in the number of vertices and in the number of pairs of the $\mathbb{P}\mathbb{C}$ game (thus of the $\mathbb{P}_1\mathbb{C}\mathbb{P}_2$ game),
- a parity condition with a number of priorities linear in the number of vertices and the number of pairs of the $\mathbb{P}\mathbb{C}$ game.

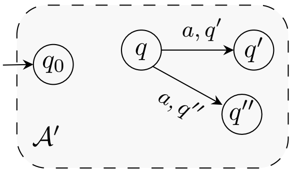

**Figure 8** The DPA $\mathcal{A}'$ used for the PSPACE-hardness of Proposition 15.

Finally, the constructed parity game can be solved in time $n^{\mathcal{O}(log(d))}$ where $n$ is its number of vertices and $d$ its number of priorities [18]. It follows that the initial NE existence problem can be decided in time exponential in $|V|$, $\Pi_{i\in\mathcal{P}}|\mathcal{A}_i|$, and $\Sigma_{i\in\mathcal{P}}d_i$. ◀

**Proof of Theorem 5, hardness and particular case of one-player games.** We are going to prove that the NE existence problem is PSPACE-complete for one-player games. In this way, we get the hardness of Theorem 5.

Before we need to study a deeply connected problem: the problem of deciding the existence of a maximal element for a preference relation $\prec$ (we also consider the existence of a minimal element).

▶ **Proposition 15.** *The problem of deciding whether an $\omega$-automatic relation $\prec$, accepted by a DPA $\mathcal{A}$, has a maximal (resp. minimal) element is* PSPACE*-complete.*

**Proof.** We mainly focus on the existence of a maximal element. We briefly discuss the existence of a minimal element at the end of the proof, as the arguments are similar. We begin with the PSPACE-membership, by studying the non-existence of a maximal element, i.e., $\forall x\ \exists y,\ x \prec y$. The algorithm constructs a nondeterministic parity automaton $\mathcal{B}$ by taking the projection of $\mathcal{A}$ to the first component of its labels. This automaton is of polynomial size and accepts the set $\{x \in \Sigma^\omega \mid \exists y \in \Sigma^\omega, x \prec y\}$. Then, one checks whether $\mathcal{B}$ is universal, with an algorithm in PSPACE [36].

Let us shift to the PSPACE-hardness. We use a reduction from the non-universality problem of nondeterministic Büchi automata (NBAs) which is PSPACE-complete [36]. Given an NBA $\mathcal{A} = (Q, \Sigma, q_0, \delta, F)$, we construct a DPA $\mathcal{A}'$, depicted in Figure 8, as follows:

- $\mathcal{A}'$ has the same states and initial state as $\mathcal{A}$,
- its alphabet $\Sigma'$ is equal to $Q \cup \Sigma$, where $Q$ and $\Sigma$ are supposed disjoint,
- its transition function $\delta'$ is defined as $q' = \delta'(q, (a, q'))$ whenever $q' \in \delta(q, a)$,
- its priority function $\alpha$ uses two priorities as follows: $\alpha(q) = 2$ if $q \in F$, otherwise $\alpha(q) = 1$.

As $\delta'$ is a function by construction, $\mathcal{A}'$ is deterministic. We denote by $\prec'$ the relation accepted by $\mathcal{A}'$. By construction, we have $x \prec' y$ if and only if $x \in \Sigma^\omega$, $y \in Q^\omega$, and $y$ is an accepting run of $\mathcal{A}$ labeled by $x$. Therefore, $x \in \Sigma^\omega$ is a maximal element if and only if $x \notin \mathcal{L}(\mathcal{A})$. We thus get the correctness of the polynomial reduction.

This completes the proof of Proposition 15 for the existence of a maximal element. Let us finally comment on the modifications needed to decide the existence of a minimal element. For the membership result, as the non-existence of a minimal element means $\forall x \exists y, y \prec x$, we have to consider the projection of $\mathcal{A}'$ on the second component of the labels in step 3. For the hardness result, we have to swap the components of the labels $(a, q)$ of the transitions of the automaton $\mathcal{A}'$. ◀

**Proof of Theorem 5 for one-player games.** Let us first present a PSPACE algorithm for the membership. Intuitively, given a one-player game $\mathcal{G} = (A, \prec)$ and an initial vertex $v_0$,

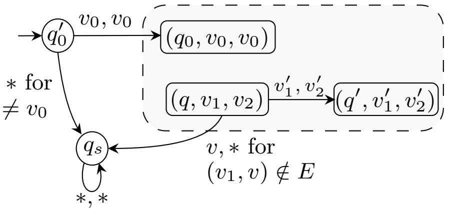

**Figure 9** The DPA $\mathcal{A}'$ for the PSPACE membership of Theorem 5 for one-player games.

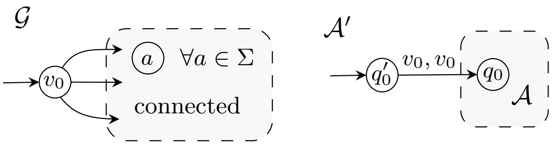

**Figure 10** The reduction used for the PSPACE-hardness of Theorem 5.

we will show that a play $\pi$ is an NE outcome from $v_0$ if and only if it is maximal for $\prec'$, a well chosen preference relation constructed from $\prec$. Hence, by Proposition 15, we will get that the existence of an NE is in PSPACE.

We define the relation $\prec'$ as follows: for all $x, y \in V^\omega$, we have $x \prec' y$ if and only if either $x, y$ are both plays starting with $v_0$ such that $x \prec y$, or $x$ is not a play starting with $v_0$. Clearly, if $\pi$ is an NE outcome from $v_0$, then it is maximal for $\prec'$. Conversely, if $x$ is maximal for $\prec'$, then $x$ is a play starting with $v_0$ and it is thus an NE outcome as $x$ is maximal.

It remains to show that $\prec'$ is accepted by some DPA $\mathcal{A}'$. This automaton, illustrated in Figure 9, is defined from the arena $A$ and the DPA $\mathcal{A}$ accepting $\prec$ in the following way. In addition to two particular states $q'_0$ and $q_s$ with priority 0, any state of $\mathcal{A}'$ is of the form $(q, v_1, v_2)$ with $q$ a state of $\mathcal{A}$ and $v_1, v_2 \in V$, and has the same priority as $q$ in $\mathcal{A}$. The state $q'_0$ is the initial state. There is a transition from $(q, v_1, v_2)$ to $(q', v'_1, v'_2)$ labeled with $(v'_1, v'_2)$ if $(v_i, v'_i) \in E$ for $i \in \{1, 2\}$ and $(q, (v'_1, v'_2), q')$ is a transition of $\mathcal{A}$. In addition, from a state $(q, v_1, v_2)$, a transition reading a label $(v'_1, v'_2)$ such that $(v_1, v'_1) \notin E$ leads to the state $q_s$ with a self loop labeled by any $(v, v') \in V \times V$. Finally, there is a transition from $q_0$ to $(q_0, v_0, v_0)$ with label $(v_0, v_0)$ (where $q_0$ is the initial state of $\mathcal{A}$), and to $q_s$ for any label $(v, v')$ such that $v \neq v_0$.

We now prove that the NE existence problem is PSPACE-hard for one-player games, with a reduction from the existence of a maximal element in an $\omega$-automatic relation, a PSPACE-complete problem by Proposition 15. The reduction works as follows. Let $\prec$ be an $\omega$-automatic relation on $\Sigma$ accepted by a DPA $\mathcal{A}$. We construct a one-player game $\mathcal{G} = (A, \prec')$ depicted in Figure 10 as follows. The set of vertices of $A$ is $\Sigma' = \Sigma \cup \{v_0\}$ with a new initial vertex $v_0$; there is an edge between every pair of letters $a, b \in \Sigma$ and an edge between $v_0$ and each letter $a \in \Sigma$. The preference relation $\prec'$ is the one accepted by the DPA $\mathcal{A}'$ of Figure 10 where the initial state $q'_0$ replaces the initial state $q_0$ of $\mathcal{A}$ and its priority function is the one of $\mathcal{A}$ extended to $q_0$ with priority 1.

It remains to show that the reduction is correct. Suppose that $\pi = v_0 x$ is an NE outcome starting in $v_0$ in $\mathcal{G}$. As it is a one-player game, $\pi$ is a maximal play with respect to $\prec'$ among all plays starting in $v_0$. As those plays all belong to $v_0\Sigma^\omega$, it follows that $x$ is a maximal element in $\prec$, by definition of $\prec'$. The other implication is proved similarly: given a maximal element $x$ in $\prec$, we get a play $\pi = v_0 x$ that is maximal in $\prec'$ among all plays starting in $v_0$, thus an NE outcome. ◀

## D Constrained NE Existence Problem

▶ **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).*

**Sketch of proof of Theorem 6, membership.** From the given game $\mathcal{G}$ and constraints $\pi_i = \mu_i(\nu_i)^\omega$ for each player $i$, we construct the same $\mathbb{P}_1\mathbb{C}\mathbb{P}_2$ game as for the NE existence problem. The only difference is on the winning condition $W_{\mathbb{P}_1\mathbb{P}_2} = W_{acc} \cup W_{dev}$ where $W_{acc}$ is modified in a way to take into account the constraints:

$$W_{acc} = \{\pi \in \mathsf{Plays}(\mathbb{P}_1\mathbb{C}\mathbb{P}_2(\mathcal{G})) \mid \mathsf{dev}(\pi) = \bot \text{ and } \pi_i \prec_i \mathsf{proj}_{V,1}(\pi),\ \forall i\}$$

This modification has an impact on the Rabin condition encoding $W_{\mathbb{P}_1\mathbb{P}_2}$. To translate $W_{acc}$ into a Rabin condition, we proceed as follows. For each $i$, we construct a DPA $\mathcal{A}'_i$ accepting the set $\{x \in V^\omega \mid \mu_i(\nu_i)^\omega \prec_i x\}$. It has $|\mathcal{A}_i| \cdot |\pi_i|$ states and $d_i$ priorities. We then construct the product $\mathcal{A}'$ of all those automata $\mathcal{A}'_i$ that accepts the set $\{x \in V^\omega \mid \mu_i(\nu_i)^\omega \prec_i x,\ \forall i\}$. This a generalized DPA whose condition can be translated into a Streett condition with $d = \Sigma_{i \in \mathcal{P}} d_i$ pairs [36]. The latter automaton is equivalent to a deterministic automaton $\mathcal{B}$ with $\mathcal{O}(|\mathcal{A}'| \cdot 2^{d \log(d)})$ states and $d$ Rabin pairs [46]. Finally, we replace the arena of the $\mathbb{P}_1\mathbb{C}\mathbb{P}_2$ game by its product with the automaton $\mathcal{B}$. Thanks to the previous argument, $W_{acc}$ is encoded as a Rabin condition on the modified arena with $d$ pairs (without forgetting the condition $\mathsf{dev}(\pi) = \bot$). As in the proof of the membership result of Theorem 5, $W_{dev}$ is translated into a Rabin condition with $d$ pairs (step 1), and the rest of the proof (steps 2 and 3) is then the same. Nevertheless, the complexity of the constrained existence problem is different as the number of vertices of the $\mathbb{P}_1\mathbb{C}\mathbb{P}_2$ game now also depends on $|\mathcal{B}|$. It follows that the constrained NE existence problem is exponential in $|V|$, $\Pi_{i \in \mathcal{P}}|\mathcal{A}_i|$, $\Pi_{i \in \mathcal{P}}|\pi_i|$, and doubly exponential in $\Sigma_{i \in \mathcal{P}} d_i$. ◀

To obtain the PSPACE-hardness result of Theorem 6, it it enough to prove it for one-player games.

**Proof of Theorem 6, one-player games.** Let us start with the PSPACE-membership of the NE constrained problem for one-player games. Given a $\mathcal{G} = (A, \prec)$ with a single preference relation, an initial vertex $v_0$, and a constraint given by a lasso $\pi$, we have to decide the existence of an NE $\sigma$ from $v_0$ in $\mathcal{G}$ such that $\pi \prec \langle \sigma \rangle_{v_0}$.

First, we define a relation $\prec'$ from $\prec$ as we did for the membership proof of Theorem 5 for one-player games in Appendix C. Recall that a play $\pi$ is an NE outcome from $v_0$ if and only if it is maximal for $\prec'$. Recall also that $\prec'$ was accepted by a DPA of size $|\mathcal{A}| \cdot |V|^2$. Second, we define an automaton $\mathcal{B}$ such that $\mathcal{L}(\mathcal{B}) = \{\rho \in \mathsf{Plays} \mid \pi \prec \rho\}$, i.e., $\mathcal{B}$ is a DPA of size $|\mathcal{A}| \cdot |\pi|$.

Our goal is then to find a maximal element of $\prec'$ in $\mathcal{B}$, i.e., to satisfy

$$\exists x \forall y,\ x \not\prec' y \wedge x \in \mathcal{L}(\mathcal{B}).$$

As PSPACE is closed under complementation, we can equivalently study the negation of this property:

$$\forall x \exists y,\ x \prec' y \vee x \notin \mathcal{L}(\mathcal{B}). \tag{2}$$

The set $\{(x, y) \mid x \prec' y \vee x \notin \mathcal{L}(\mathcal{B})\}$ is accepted by a generalized DPA with a disjunction of two parity conditions, i.e., by a deterministic Streett automaton. This automaton has size $|V|^2 \cdot |\mathcal{A}|^2 \cdot |\pi|$. Therefore, the set $\{x \mid \exists y,\ x \prec' y \vee x \notin \mathcal{L}(\mathcal{B})\}$ is accepted by a nondeterministic Streett obtained from the previous one by projection on the first component. To check (2), it remains to perform a universality check on the latter automaton, which is done in PSPACE [7].

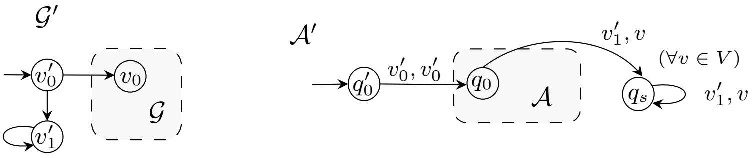

**Figure 11** The reduction used for the PSPACE-hardness of Theorem 6.

Let us now prove the PSPACE-hardness. We use a reduction from the NE existence problem and suppose that there is only one player, as the NE existence problem is already PSPACE-hard in this case. Given a game $\mathcal{G} = (A, \prec)$ with only player 1 with a preference relation $\prec$ accepted by a DPA $\mathcal{A}$, and $v_0$ an initial vertex, we construct a new game $\mathcal{G}' = (A', \prec')$ as depicted in Figure 11. Its arena $A'$ has $V' = V \cup \{v'_0, v'_1\}$ as set of vertices, with $v'_0$ and $v'_1$, two new vertices owned by player 1. It is a copy of $A$ with $v'_0$, the new initial vertex, having $v_0$ and $v'_1$ as successors, and $v'_1$ having itself as successor. The preference relation $\prec' \subseteq (V')^\omega \times (V')^\omega$ is the one accepted by the DPA $\mathcal{A}'$ of Figure 11 where the initial state $q'_0$ replaces the initial state $q_0$ of $\mathcal{A}$ and its priority function is the one of $\mathcal{A}$ extended to $q'_0$ and $q_s$, both with priority 0. In addition to the transitions of $\mathcal{A}$, we have a transition from $q'_0$ to $q_0$ labeled by $(v'_0, v'_0)$ and for all $v \in V$, a transition from $q_0$ to $q_s$ (resp. from $q_s$ to $q_s$) labeled by $(v'_1, v)$. Note that, thanks to the sink state $q_s$,

$$v'_0(v'_1)^\omega \prec' v'_0 x \text{ for all } x \in V^\omega. \tag{3}$$

We impose a constraint given by the lasso $v'_0(v'_1)^\omega$.

Let us show that there is an NE in $\mathcal{G}$ from $v_0$ if and only if there is one in $\mathcal{G}'$ from $v'_0$ whose outcome $\pi$ satisfies $v'_0(v'_1)^\omega \prec' \pi$ (Recall that NEs are composed of a single strategy, as there is only one player). Let us first suppose that there exists an NE $\sigma$ in $\mathcal{G}$. Let $\sigma'$ be a strategy in $\mathcal{G}'$ defined as $\sigma'(v'_0) = v_0$ and $\sigma'(v'_0 h) = \sigma(h)$ for any history $h$. Clearly, by construction of $\prec'$, see (3), we have $v'_0(v'_1)^\omega \prec' \langle \sigma' \rangle_{v'_0}$. Let us explain why the profile $\sigma'$ is an NE. The deviation $v'_0(v'_1)^\omega$ is not profitable for player 1, by (3). Any other deviation is necessarily of the form $v'_0\pi$ with $\pi$ a deviation in $\mathcal{G}$ with respect to $\sigma$. Hence, it cannot be profitable as $\sigma$ is an NE and by definition of $\prec'$. Conversely, suppose that there exists an NE $\sigma'$ in $\mathcal{G}'$ such that $v'_0(v'_1)^\omega \prec' \pi' = \langle \sigma' \rangle_{v'_0}$. Note that $\pi' \neq v'_0(v'_1)^\omega$ by (3) and as $\sigma'$ is an NE. Therefore, we can define a strategy profile $\sigma$ in $\mathcal{G}$ such that $\sigma(h) = \sigma'(v'_0 h)$ for any history $h$. This is an NE in $\mathcal{G}$ since $\sigma'$ is an NE in $\mathcal{G}'$.

This shows that the constrained NE existence problem is PSPACE-hard for the single-player case. ◀

## E Proof of Proposition 9

For the proof of Proposition 9, we use logspace reductions to show the NL-hardness results. Such a reduction consists in computing a polynomially bounded function $f$ with a deterministic Turing Machine using three tapes: a read-only input tape containing the input $x$ of length $n$, a write-only output tape that will contain $f(x)$ at the end of the execution, and a read-write work tape with $\log(n)$ cells. The reader can consult [1, 48] for more details about logspace reductions. Let us also recall that the complexity class NL is closed under complementation [48]. Before providing the proof of Proposition 9, we need the following result on generalized nondeterministic parity automata (NPAs) with a positive Boolean combination of a constant number of parity conditions.

▶ **Proposition 16.** *The problem of deciding whether an $\omega$-regular language $L \subseteq \Sigma^\omega$ is not empty (resp. universal) is* NL*-complete if $L$ is accepted by a generalized NPA (resp. DPA) with a constant number of parity conditions.*

**Proof.** Let us show the NL-membership for the non-emptiness problem for both statements. Consider a generalized NPA $\mathcal{A}$ with a positive Boolean combination of $d$ parity conditions, with $d$ constant. Since parity conditions only handle infinite occurrences of states of a run, by [9, Proposition 3.1], we know that if there is a word $w$ accepted by $\mathcal{A}$, then there exists a lasso $\mu(\nu)^\omega$ accepted by $\mathcal{A}$ such that $|\mu|, |\nu| \leq |Q|^2$, where $|Q|$ is the number of states of $\mathcal{A}$. Hence, we can guess the length $n \leq 2|Q|^2$ of such a lasso, and the lasso itself on the fly, state by state. To check whether the guessed lasso is accepted by $\mathcal{A}$, we retain the maximum priority occurring in $\nu$ for each of the $d$ parity conditions, in a way to verify whether the Boolean combination of those conditions is true or not. At any time, the amount of information to be stored is logarithmic since a state and $d$ priorities take logarithmic space when written in binary (recall that $d$ is constant).

For the universality problem in case $\mathcal{A}$ is deterministic, it amounts to solve the non-emptiness problem for the complement automaton $\mathcal{A}'$. The latter automaton is constructed from $\mathcal{A}$ where in its Boolean combination, each $\vee$ (resp. $\wedge$) is replaced by $\wedge$ (resp. $\vee$), and the priorities are all incremented by one in each of its parity conditions. So, the size of $\mathcal{A}'$ is the same and the determinism is preserved. Hence, we can perform the same algorithm as above.

Since the non-emptiness for NBAs and the universality problems for DBAs are both NL-complete [36], and a Büchi condition is a special case of a generalized parity condition, we get the NL-hardness of every statement. ◀

We now proceed to the proof of Proposition 9. We denote by $\mathcal{A}$ a DPA accepting an $\omega$-automatic relation $R \subseteq \Sigma^\omega \times \Sigma^\omega$.

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

**Proof of Proposition 9 - reflexivity.** Let us start with the NL-membership. From $\mathcal{A}$, we construct a new automaton $\mathcal{A}'$ over $\Sigma$ by first erasing all transitions labeled by $(a, b)$ with $a \neq b$, and then replacing each label $(a, a)$ of the remaining transitions by $a$. Clearly, $\mathcal{A}'$ is a DPA and we get that $\{(x, x) \mid x \in \Sigma^\omega\} \subseteq \mathcal{L}(\mathcal{A})$ if and only if $\mathcal{L}(\mathcal{A}') = \Sigma^\omega$. Therefore, testing whether $R$ is reflexive reduces to checking the universality of $\mathcal{A}'$. The latter problem is NL for DPAs by Proposition 16. Note that $\mathcal{A}'$ is constructed on the fly from $\mathcal{A}$ while guessing a lasso not accepted by $\mathcal{A}'$. This requires to use pointers on states and transitions of $\mathcal{A}$, those pointers being stored in a logarithmic space of the work tape.

Let us continue with the NL-hardness. We proceed by reduction from the universality problem for DBAs which is NL-complete [36]. Given such an automaton $\mathcal{A}$ over $\Sigma$, we construct a new automaton $\mathcal{A}' = \mathcal{A} \times \mathcal{A}$ that accepts the relation $R = \mathcal{L}(A) \times \mathcal{L}(A)$ over $\Sigma \times \Sigma$. The automaton $\mathcal{A}'$ has a size $|\mathcal{A}|^2$ and is a generalized DBA with a conjunction of two Büchi conditions, and can thus be transformed into a DBA $\mathcal{B}$ of size $2|\mathcal{A}'|^2$ [36]. Moreover, we have that for all $x \in \Sigma^\omega$, $x \in \mathcal{L}(\mathcal{A})$ if and only if $(x, x) \in \mathcal{L}(\mathcal{B})$. That is, $\mathcal{L}(\mathcal{A}) = \Sigma^\omega$ if and only if $R$ is reflexive. This establishes the correctness of the reduction. Finally, it is a logspace reduction. Indeed, we need to use pointers on the input tape, stored in the work tape, in a way to iterate on the states and transitions of $\mathcal{A}$, and construct step by step the states and transitions of $\mathcal{B}$ on the output tape; we also need to store an extra bit to remember the copy of $\mathcal{A}'$ (to get the DBA $\mathcal{B}$). ◀

**Proof of Proposition 9 - irreflexivity.** For irreflexivity, note that $R$ is irreflexive if and only if $(\Sigma^\omega \times \Sigma^\omega)\backslash R$ is reflexive. So, as the complement of a DPA is still a DPA of the same size, we can repeat the same NL algorithm as above with $(\Sigma^\omega \times \Sigma^\omega)\backslash R$, and a reduction from deciding the reflexivity of a relation to obtain the NL-hardness. ◀

**Proof of Proposition 9 - transitivity.** We begin with the NL-membership. The relation $R$ is not transitive if there exist $x, y, z \in \Sigma^\omega$ such that $(x, y) \in R$ and $(y, z) \in R$ but $(x, z) \notin R$. Testing whether $R$ is not transitive thus reduces to testing whether the language

$$L = \{(x, y, z) \in (\Sigma^\omega)^3 \mid (x, y) \in R \wedge (y, z) \in R \wedge (x, z) \notin R\}$$

is not empty. We can construct an automaton $\mathcal{A}'$ accepting $L$ which is a generalized DPA with a conjunction of three parity conditions. This automaton has a polynomial size (the complementation of $\mathcal{A}$ leads to an automaton with the same size as it suffices to increment each priority by 1). Hence, we can check whether $L$ is not empty with the NL algorithm of Proposition 16. Note that $\mathcal{A}'$ is constructed on the fly while guessing a lasso accepted by $\mathcal{A}'$.

We now prove the NL-hardness. We use a reduction from the universality problem of DBAs which is NL-complete [36]. Given a DBA $\mathcal{A}$ over the alphabet $\Sigma$, we construct the relation $R = R_1 \cup R_2 \cup R_3 \cup R_4$ on $(\Sigma')^\omega \times (\Sigma')^\omega$ with $\Sigma' = \Sigma \cup \{\#\}$ such that:

- $R_1 = \Sigma^\omega \times \{\#^\omega\}$,
- $R_2 = \{\#^\omega\} \times \Sigma^\omega$,
- $R_3 = \mathcal{L}(\mathcal{A}) \times \mathcal{L}(\mathcal{A})$,
- $R_4 = \{(\#^\omega, \#^\omega)\}$.

Let us show the correctness of the reduction. Suppose that $\mathcal{L}(\mathcal{A}) = \Sigma^\omega$, it means that $R_3 = \Sigma^\omega \times \Sigma^\omega$. Let $x, y, z \in (\Sigma')^\omega$ such that $(x, y) \in R$ and $(y, z) \in R$, and let us show that $(x, z) \in R$. If $x = \#^\omega$, then $y \in \Sigma^\omega$ or $y = \#^\omega$, so $(x, z) \in R_2 \cup R_4$. Otherwise $x \neq \#^\omega$, thus by $R_3 = \Sigma^\omega \times \Sigma^\omega$ and $R_1$, we have $(x, z) \in R_1 \cup R_3$. Conversely, if $\mathcal{L}(\mathcal{A}) \neq \Sigma^\omega$, then there exists $x \in \Sigma^\omega\backslash\mathcal{L}(\mathcal{A})$. So, we have $(x, \#^\omega) \in R_1$, $(\#^\omega, x) \in R_2$ but $(x, x) \notin R$. Hence, $\mathcal{L}(\mathcal{A}) = \Sigma^\omega$ if and only if $R$ is transitive. Furthermore, $R$ is accepted by a DBA $\mathcal{A}'$ that we construct as follows. Each $R_i$, $i \neq 3$, is trivially accepted by a DBA of constant size. For $R_3$, we construct the automaton $\mathcal{A} \times \mathcal{A}$ which is a generalized DBA with a conjunction of two Büchi conditions, that can be transformed into a DBA. We then create a new initial state $q_0$ (replacing the initial state $q_0^i$ for each $R_i$) and duplicate the transitions outgoing[16] each $q_0^i$ as a transition outgoing $q_0$. In this way we get a DBA $\mathcal{A}'$ accepting $R$ of polynomial size. One can check that this reduction is a logspace reduction. ◀

**Proof of Proposition 9 - $\neg$-transitivity.** Note that $R$ is $\neg$-transitive if and only if $(\Sigma^\omega \times \Sigma^\omega)\backslash R$ is transitive. Hence, as the complement of a DPA is still a DPA of the same size, we can repeat the same NL algorithm as for transitivity with $(\Sigma^\omega \times \Sigma^\omega)\backslash R$, and a reduction from decide the transitivity of a relation to obtain the NL-hardness. ◀

**Proof of Proposition 9 - totality.** We begin with the NL-membership. Recall that $R$ is total if for all $x, y \in \Sigma^\omega$, we have $(x = y) \vee ((x, y) \in R) \vee ((y, x) \in R)$. Equivalently, $R$ is total if $R' = R_= \cup R \cup \tilde{R}$ is equal to $\Sigma^\omega \times \Sigma^\omega$ such that $R_= = \{(x, x) \mid x \in \Sigma^\omega\}$ and $\tilde{R} = \{(y, x) \mid (x, y) \in R\}$. From $\mathcal{A}$, it is easy to construct a generalized DPA $\mathcal{A}'$ of polynomial size with a disjunction of three parity conditions which accepts $R'$. We then run the NL algorithm of Proposition 16 to check whether $\mathcal{A}'$ is universal while constructing $\mathcal{A}'$ on the fly.

[16] of the form $(a, \#)$ (resp. $(\#, a), (a, b), (\#, \#)$) for $R_1$ (resp. $R_2, R_3, R_4$), with $a, b \in \Sigma$.

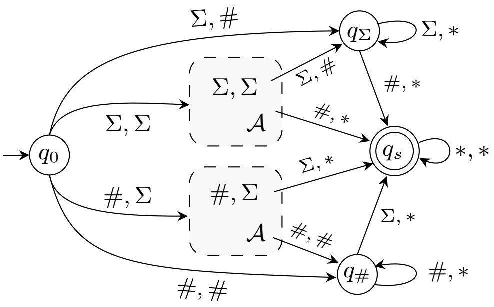

**Figure 12** The DBA $\mathcal{A}'$ used in the reduction for the totality property of Proposition 9.

Let us turn to the NL-hardness. We use a reduction from the universality problem of DBAs which is NL-complete [36]. From a DBA $\mathcal{A}$ over the alphabet $\Sigma$, we define the alphabet $\Sigma' = \Sigma \cup \{\#\}$ and the relation $R = R_1 \cup R_2$ where:

- $R_1 = (\Sigma^\omega \cup \{\#^\omega\}) \times \mathcal{L}(\mathcal{A})$ and
- $R_2 = L \times (\Sigma')^\omega$, with $L = (\Sigma')^\omega \backslash (\Sigma^\omega \cup \{\#^\omega\})$.

Let us prove the correctness of the reduction. Suppose that $\mathcal{L}(\mathcal{A}) \neq \Sigma^\omega$. Given $x \in \Sigma^\omega \backslash \mathcal{L}(\mathcal{A})$, we have $x \neq \#^\omega$, $(\#^\omega, x) \notin R$, and $(x, \#^\omega) \notin R$. Hence, $R$ is not total. Suppose now that $R$ is not total, i.e., there exists $x \neq y$ such that $(x,y),(y,x) \notin R$. By definition of $R_2$, we must have $x, y \in \Sigma^\omega \cup \{\#^\omega\}$. Since $x \neq y$, we know that $x$ or $y$ is in $\Sigma^\omega$, w.l.o.g., let us say $y \in \Sigma^\omega$. As $(x,y) \notin R_1$, we get $y \notin \mathcal{L}(\mathcal{A})$. Therefore $\mathcal{L}(\mathcal{A}) = \Sigma^\omega$ if and only if $R$ is total.

Moreover, $R$ is accepted by the DBA $\mathcal{A}'$ depicted in Figure 12. Let us give some explanations. W.l.o.g., we suppose that the automaton $\mathcal{A}$ is complete. The part of $\mathcal{A}'$ composed of the initial state $q_0$ and the two copies of $\mathcal{A}$ accepts the relation $R_1 = (\Sigma^\omega \times \mathcal{L}(\mathcal{A})) \cup (\{\#^\omega\} \times \mathcal{L}(\mathcal{A}))$. The relation $R_2$ is accepted thanks to the accepting sink state $q_s$. Note that $L$ is composed of all words containing at least one symbol of $\Sigma$ and one symbol $\#$, a condition that has to be satisfied to reach $q_s$. The states $q_\Sigma$ and $q_\#$ retain the information that it is impossible to have $(x,y) \in R_1$ due to the symbol $\#$ seen inside $y$, implying $y \notin \mathcal{L}(\mathcal{A})$. Finally, note that $\mathcal{A}'$ is deterministic, thus a DBA and in particular a DPA. This completes the proof as the proposed reduction is a logspace reduction. ◀

## F Adaptation of Reductions for Preorders

In this section, we present how to adapt the hardness proofs of Theorems 3–6 for $\omega$-automatic preorders (instead of $\omega$-automatic relations). For every proof, the general idea is to modify each DPA used in the reductions so that it accepts a preorder $\precsim$ extending the relation $\prec$ initially accepted (i.e., in the sense that $x \prec y$ if and only if $x \precsim y$ and $y \not\precsim x$).

**NE Checking Problem.** For the PSPACE-hardness of Theorem 3 presented in Appendix A, relations $\prec_i$ are empty for $1 \leq i \leq n$, while $\prec_{n+1} = L^c \times L$. The only small modification is to define $\precsim_i = V^\omega \times V^\omega$, for $1 \leq i \leq n$, and $\precsim_{n+1} = (L^c \times L) \cup (L^c \times L^c) \cup (L \times L)$. All these relations are now reflexive and transitive. However, some explanation is necessary for the transitivity of $\precsim_{n+1}$. Let $x, y, z \in V^\omega$ be such that $x \precsim_{n+1} y$ and $y \precsim_{n+1} z$. If $y \in L^c$, we must have $x \in L^c$ so $x \precsim_{n+1} z$. Otherwise, $y \in L$, then we must have $z \in L$ so we also have $x \precsim_{n+1} z$. As $L$ and $L^c$ are DBAs, the cartesian product of two DBAs is a generalized DBA with a conjunction of two Büchi conditions, thus a DBA [36], and the union of two DBAs is still a DBA, we can conclude because a DBA is in particular a DPA.

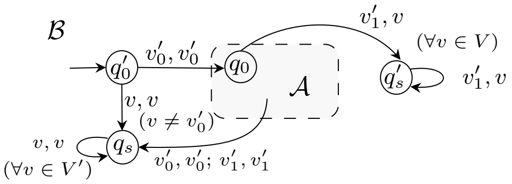

**Figure 13** The modified automaton $\mathcal{B}$ for the hardness of Theorem 6 in the case of preorders.

**NE Outcome Checking Problem.** For the Parity-hardness of Theorem 4, the empty relation $\prec_2$ is transformed into $\precsim_2 = (V')^\omega \times (V')^\omega$ (recall that $V' = V \cup \{v_0'\}$ for $v_0'$ the new initial state). We then modify the DPA $\mathcal{A}_1$ for $\prec_1$ of Figure 4 by adding a new state $q_s$ with priority 0 and new transitions $(q_0, (v, v), q_s)$ for all $v \in V$, and $(q_s, (v, v), q_s)$ for all $v \in V'$. The modified DPA accepts $\precsim_1$, the reflexive closure of $\prec_1$. One can easily check that it is transitive.

**NE Existence Problem.** To show the PSPACE-hardness of Theorem 5 for preorders, we need to show the PSPACE-hardness of Proposition 15 for preorders (see Appendix C).

Let us consider the relation $\prec'$ in the hardness proof of Proposition 15 and the DPA $\mathcal{A}'$ accepting it. We modify this automaton into a DPA $\mathcal{B}$ accepting $\precsim'$. We add to $\mathcal{A}'$ two extra states:

- a sink state $q_s$ with a loop labeled $(b, b)$ for each $b \in Q \cup \Sigma$,
- a state $q_0'$ that becomes the new initial state, with the transitions $\delta'(q_0', (a, q)) = \delta'(q_0, (a, q))$ whenever the transition exists from $q_0$, and $\delta'(q_0', (b, b)) = q_s$ for all $b \in Q \cup \Sigma$.

The modified automaton is still deterministic. The priority function $\alpha$ is extended such that $\alpha(q_0') = 1$ and $\alpha(q_s) = 2$. The relation accepted by $\mathcal{B}$ is clearly reflexive (thanks to the sink state $q_s$). Let us show that it is also transitive: let $x, y, z \in (\Sigma')^\omega$ be such that $(x, y)$ and $(y, z) \in R$. As $(t, t') \in R$ with $t \neq t'$ implies that $t \in \Sigma^\omega$ and $t' \in Q^\omega$, we must have $x = y$ or $y = z$. It follows that the relation of $\mathcal{B}$ is transitive, and thus it is an $\omega$-automatic preorder $\precsim$.

Now that Proposition 15 holds for preorders, let us modify the reduction of Theorem 5 presented in Appendix C. In Figure 10, we can assume that $\mathcal{A}$ is complete. Moreover, we add an extra state $q_s$ with a priority 0 and transitions $(q_0', (a, a), q_s)$ for all $a \in \Sigma$, $(q_s, (a, a), q_s)$ for all $a \in \Sigma \cup \{v_0\}$, and $(q, (v_0, v_0), q_s)$ for all $q \notin \{q_0, q_s\}$. One can easily verify that the new relation $\precsim'$ is reflexive. Let us prove that it is transitive. Let $x \precsim' y$ and $y \precsim' z$, and let us discuss the following three cases:

- If $y$ starts with some symbol $a \in \Sigma$, then $x = y = z$ and $x \precsim' z$.
- If $y$ is of the form $v_0 y'$ with $y' \in \Sigma^\omega$, then it is also the case for $x$ and $z$. As $\precsim$ is transitive, we get that $x \precsim' z$.
- If $y$ is of the form $v_0 w_y v_0 y'$ with $w_y \in \Sigma^*$, then $x = v_0 w_x v_0 x'$ and $z = v_0 w_z v_0 z'$ with $w_x, w_z \in \Sigma^*$ and $|w_x| = |w_y| = |w_z|$. Therefore, as $\mathcal{A}$ is complete, we get $x \precsim' z$.

**Constrained NE Existence Problem.** For the hardness of Theorem 6, we have to modify the DPA $\mathcal{A}'$ in Figure 11 into a DPA $\mathcal{B}$ to make it reflexive in the following way (see Figure 13). We add a new sink state $q_s'$ with a priority 0. We add transitions $(q_0', (v, v), q_s')$ for all $v \neq v_0'$, $(q_s', (v, v), q_s')$ for all $v \neq v_0'$, and $(q, (v_0', v_0'), q_s')$, $(q, (v_1', v_1'), q_s')$ for all state $q$ in the copy of $\mathcal{A}$.

Let us prove that the relation $\precsim'$ accepted by the $\mathcal{B}$ is transitive (w.l.o.g., we suppose that $\mathcal{A}$ is complete). Let $x \precsim' y$ and $y \precsim' z$. Notice that $x \in v_0'V^\omega \Rightarrow y \in v_0'V^\omega$ and $y \in v_0'V^\omega \Rightarrow z \in v_0'V^\omega$ (due to the copy of $\mathcal{A}$ inside $\mathcal{B}$). Let us discuss the following cases:

- If $y$ does not start with $v_0'$, then $x = y = z$ and $x \precsim' z$. Otherwise, $y$ starts with $v_0'$, thus also $x$ and $z$.
- If $x \in v_0'V^\omega$, then $y, z \in v_0'V^\omega$ and the runs with label $(x, y)$ and $(y, z)$ both stay in $\mathcal{A}_i$. As $\precsim$ is transitive, we get that $x \precsim' z$.
- If $x \notin v_0'V^\omega$ and $y \in v_0'V^\omega$ (and thus $z \in v_0'V^\omega$), then $x = v_0'(v_1')^\omega$ and then the run labeled by $(x, z)$ ultimately loops in $q_s'$, showing that $x \precsim' z$.
- If $x, y \notin v_0'V^\omega$ and $z \in v_0'V^\omega$, then necessarily $x = y = v_0'(v_1')^\omega$, and we get $x \precsim' z$.
- If $x, y, z \notin v_0'V^\omega$, then $x$ (resp. $y$, $z$) is the form $v_0'w_xv_i'x'$ (resp. $v_0'w_yv_i'y'$, $v_0'w_zv_i'z'$) with $w_x, w_y, w_z \in V^*$ such that $|w_x| = |w_y| = |w_z|$, and $i \in \{0, 1\}$. Therefore, as $\mathcal{A}$ is complete, we get $x \precsim' z$.

## G Proof of Proposition 10

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

**Proof.** We use [39, Lemma 3] to prove this proposition. Let $E_1 = \{(x, y) \mid \forall z, x \precsim z \text{ iff } y \precsim z\}$ and $E_2 = \{((x_1, x_2), (y_1, y_2)) \mid x_1 \precsim x_2 \text{ iff } y_1 \precsim y_2\}$ be two equivalence relations. From [39, Lemma 3], we know that $\precsim$ is $\omega$-recognizable if and only if both $E_1$ and $E_2$ have finite index. Note that by definition, $E_2$ has at most two equivalence classes. Hence, to establish Proposition 10, we will show that $E_1 = \sim$. Let $x, y \in \Sigma^\omega$. If $(x, y) \in E_1$, we can particularize $z$ to $x$ and then $y$, to get that $x \sim y$ by reflexivity of $\precsim$. If $x \sim y$, for all $z \in \Sigma^\omega$, we get by transitivity of $\precsim$ that $x \precsim z$ if and only if $y \precsim z$, thus $(x, y) \in E_1$. ◀

## H Proof of the Existence of NE with Total $\omega$-Recognizable Preorders

In this section, we assume that each game $\mathcal{G}$ has an $\omega$-recognizable preorder $\precsim_i$ for each $i \in \mathcal{P}$. We aim at proving Theorem 11 when every relation is total.

By Proposition 10, we denote by $\Lambda_i$ the finite lattice formed by the equivalence classes of $\sim_i$. We also denote by $[x]_i$ the equivalence class of the word $x$ with respect to $\sim_i$. By abusive notation, we write $[x]_i \precsim_i [y]_i$ whenever $x \precsim_i y$.

**Useful Lemmas** In addition to Proposition 10, we first mention some useful properties about preorders.

▶ **Lemma 17.** *Let $\precsim$ be an $\omega$-recognizable preorder over $\Sigma$ and $\sim$ the related equivalence relation. Then, in each equivalence class of $\sim$, there exists a lasso. Moreover, given a word $x \in \Sigma^\omega$, there exists a prefix $\mu\nu$ of $x$ such that $x \sim \mu(\nu)^\omega$.*

**Proof.** Let $x \in \Sigma^\omega$. As $\precsim$ is a preorder, we have $x \precsim x$. As $\precsim$ is $\omega$-recognizable, i.e., $\precsim$ is equal to $\cup_{i=1}^{\ell} X_i \times Y_i$ where $X_i, Y_i$ are $\omega$-regular languages, there exists $i$ such that $(x, x) \in X_i \times Y_i$. Therefore, $x$ belongs to $X_i \cap Y_i$ which is still $\omega$-regular and thus contains a lasso $\mu(\nu)^\omega$ such that $\mu\nu$ is a prefix of $x$. It follows that $(x, \mu(\nu)^\omega) \in X_i \times Y_i$ and $(\mu(\nu)^\omega, x) \in X_i \times Y_i$, thus $x \precsim \mu(\nu)^\omega$ and $\mu(\nu)^\omega \precsim x$, that is, $x \sim \mu(\nu)^\omega$. ◀

Note that Lemma 17 does not hold in the general case of $\omega$-automatic preorders. Let us take $\precsim$ defined as $x \precsim y$ if and only if $x = y$. This is an $\omega$-automatic preorder where for each word $x$, its equivalence class is the singleton $\{x\}$. There is an infinite number of such classes, showing that $\precsim$ is not $\omega$-recognizable (by Proposition 10), and Lemma 17 does not hold.

**Total Preorders** We now suppose that each preorder $\precsim_i$ is total, that is, the lattice $\Lambda_i$ is a finite *total* order. This means that the finite number of equivalence classes of $\sim_i$ are totally ordered from the lowest one to the highest one. Consequently, we have the next important property: for all $x, y \in \Sigma^\omega$,

$$x \not\precsim_i y \quad \Leftrightarrow \quad y \precsim_i x.$$

Let us state Theorem 11 in the particular setting of total preorders.

▶ **Theorem 18.** *When the preference relations of a game are all $\omega$-recognizable and total preorders, then there always exists an NE.*

To prove Theorem 18, we take inspiration from the work of [31] and [14], where the existence of NEs is studied through the concept of value and optimal strategy. In the next definitions, for a fixed $i \in \mathcal{P}$, the notation $\Sigma_i$ (resp. $\Sigma_{-i}$) is used for the set of all strategies of player $i$ (resp. player $-i$).

▶ **Definition 19.** *Let $\precsim_i$ be the preference relation of player $i$ and $-i$ be the coalition of the other players. Let $v$ be a vertex. We define the following* lower *and* upper values*:*

- $\underline{val}_i(v) = \max_{\sigma_i \in \Sigma_i} \min\{[\pi]_i \mid \pi$ *consistent with* $\sigma_i$ *and* $\mathsf{first}(\pi) = v\}$,
- $\overline{val}_i(v) = \min_{\sigma_{-i} \in \Sigma_{-i}} \max\{[\pi]_i \mid \pi$ *consistent with* $\sigma_{-i}$ *and* $\mathsf{first}(\pi) = v\}$.

Note that the lower and upper values are well defined as the lattice $\Lambda_i$ is finite.

▶ **Lemma 20.** *We have $\underline{val}_i(v) \precsim_i \overline{val}_i(v)$ for each $v \in V$.*

**Proof.** Let $\sigma_i^v$ that realizes $\underline{val}_i(v)$ and $\sigma_{-i}^v$ that realizes $\overline{val}_i(v)$. Then, we have $\underline{val}_i(v) \precsim_i [\langle \sigma_i^v, \sigma_{-i}^v \rangle_v]_i \precsim_i \overline{val}_i(v)$. ◀

▶ **Definition 21.** *If $\underline{val}_i(v) = \overline{val}_i(v)$, then we speak about the* value of $v$ *denoted by $val_i(v)$ and two strategies $\sigma_i^v \in \Sigma_i$, $\sigma_{-i}^v \in \Sigma_{-i}$ that realize the value are called* optimal.

Note that an optimal strategy $\sigma_i^v$ ensures consistent plays $\pi$ starting at $v$ such that $val_i(v) \precsim_i [\pi]_i$. This leads us to introduce the concept of *threshold game* $(A, \precsim_i, \Omega_i)$ defined from $\mathcal{G}$ and a lasso $\rho$, and with $\Omega_i = \{x \in V^\omega \mid \rho \precsim_i x\}$. It is a zero-sum player game between the players $i$ and $-i$, whose objective of player $i$ is the set $\Omega_i$ (while the objective of player $-i$ is $V^\omega \backslash \Omega_i$). Hence, if $\rho$ is a lasso in the equivalence class $val_i(v)$ such that $\mathsf{first}(\pi) = v$ (it exists by Lemma 17), $\sigma_i^v$ is nothing more than a winning strategy in this threshold game. Similarly, an optimal strategy $\sigma_{-i}^v$ ensures consistent plays $\pi$ starting at $v$ such that $[\pi]_i \precsim_i val_i(v)$. It is thus a winning strategy in the variant of threshold game $(A, \precsim_i, \Omega_{-i})$ where $\Omega_{-i} = \{x \in V^\omega \mid x \precsim_i \rho\}$ is the objective of player $-i$.

▶ **Proposition 22.** *Every threshold game (and its variant) is determined, with finite-memory winning strategies for both players.*

**Proof.** Let $(A, \precsim_i, \Omega_i)$ be a threshold game defined from $\mathcal{G}$ and a lasso $\rho$. From the DPA $\mathcal{A}_i$ accepting $\precsim_i$, we construct a DPA $\mathcal{A}_i'$ accepting $\Omega_i$, which is the product between the lasso $\rho$ and $\mathcal{A}_i$. Then, we construct a zero-sum parity game $\mathcal{H}$ from the product between the arena $A$ of $\mathcal{G}$ and $\mathcal{A}'_i$, such that its parity condition encodes the objective $\Omega_i$.[17] Clearly, the existence of a winning strategy in the threshold game is equivalent to the existence of a winning strategy in $\mathcal{H}$. As parity games are determined and have memoryless winning strategies for both players [30], this completes the proof. Note that the argument is similar for the variant with the objective $\Omega_{-i}$ for player $-i$. ◀

▶ **Proposition 23.** *There always exists a value $val_i(v)$ for each $v \in V$. Moreover, there exist finite-memory optimal strategies $\sigma_i^v$ and $\sigma_{-i}^v$ whose outcome $\langle \sigma_i^v, \sigma_{-i}^v \rangle_v$ is a lasso in the equivalence class $val_i(v)$.*

**Proof.** Let us suppose for a contradiction that $\underline{val}_i(v) \prec_i \overline{val}_i(v)$ for some $v$. It implies the existence of a play $\rho \in \overline{val}_i(v)$ with $\mathsf{first}(\rho) = v$ such that $\underline{val}_i(v) \prec_i [\rho]_i$. By Lemma 17, we can assume that $\rho$ is a lasso. In other words, we have

$$\neg(\exists \sigma_i \in \Sigma_i, \forall \sigma_{-i} \in \Sigma_{-i},\ \rho \precsim_i \langle \sigma_i, \sigma_{-i} \rangle_v). \tag{4}$$

We consider the threshold game $(A, \precsim_i, \Omega_i)$ with the objective $\Omega_i = \{x \in V^\omega \mid \rho \precsim_i x\}$ for player $i$. By determinacy, given by Proposition 22, and totality of $\precsim_i$, (4) is equivalent to

$$\exists \sigma_{-i} \in \Sigma_{-i}, \forall \sigma_i \in \Sigma_i,\ \langle \sigma_i, \sigma_{-i} \rangle_v \prec_i \rho.$$

The existence of such a strategy $\sigma_{-i}$ is in contradiction with the definition of $\overline{val}_i(v)$, since $\rho \in \overline{val}_i(v)$. Hence, for all $v$, we have $\underline{val}_i(v) = \overline{val}_i(v)$ and $val_i(v)$ exists.

Let us now prove the existence of finite-memory optimal strategies. Let $\rho$ be a lasso in the equivalence class $val_i(v)$ such that $\mathsf{first}(\rho) = v$. By Proposition 22 (where $\Omega_i$ is defined with this lasso $\rho$), there exists a (winning) optimal strategy $\sigma_i$ that is finite-memory. Similarly, there exists a finite-memory optimal strategy $\sigma_{-i}$ for player $-i$ and his objective $\Omega_{-i} = \{x \in V^\omega \mid x \precsim_i \rho\}$. ◀

Now that the values and optimal strategies are defined, we still need to introduce the concept of subgame before proceeding to the proof of Theorem 18. Given a game $\mathcal{G} = (A, (\precsim_i)_{i \in \mathcal{P}})$ and a history $h \in V^*$, we denote by $\mathcal{G}_{\upharpoonright h} = (A, (\precsim_i^h)_{i \in \mathcal{P}})$ the *subgame from* $h$, where for each player $i$, we define the relation $\precsim_i^h$ by:

$$\forall x, y \in V^\omega,\ x \precsim_i^h y \ \Leftrightarrow\ hx \precsim_i hy.$$

Hence, this relation only refers to the preference relation $\precsim_i$ restricted to words having $h$ as a prefix. We also define the relations $\prec_i^h$ and $\sim_i^h$ as expected. By definition, we have $x \sim_i^h y$ if and only if $hx \sim_i hy$. Hence, the equivalence class $[x]_i^h$ of $x$ for $\sim_i^h$ is mapped to the equivalence class $[hx]_i$ of $hx$ for $\sim_i$, called the *projection* of $[x]_i^h$.

▶ **Lemma 24.** *Given the $\omega$-recognizable total preorder $\precsim_i$ and any history $h \in V^*$, the relation $\precsim_i^h$ is also an $\omega$-recognizable total preorder.*

**Proof.** By definition, $\precsim_i^h$ is clearly a total preorder. Furthermore, we can easily define a DPA accepting $\precsim_i^h$ from the DPA $\mathcal{A}_i$ accepting $\precsim_i$. Indeed, we simply take a copy of $\mathcal{A}_i$ and replace its initial state with the unique state reached by reading the pair $(h, h)$. Finally, to show that $\precsim_i^h$ is $\omega$-recognizable, let us prove that $\sim_i^h$ has finite index (by Proposition 10). As the equivalence class of any $x$ for $\sim_i^h$ is mapped to its projection $[hx]_i$ and $\sim_i$ has a finite index, it is also the case for $\sim_i^h$. ◀

[17] Note that in the proof of the membership result of Theorem 4, such a zero-sum generalized parity game was already constructed for the objective $\{x \in V^\omega \mid \rho \prec_i x\}$.

We are now ready to prove Theorem 18.

**Proof of Theorem 18.** As done for $\mathcal{G}$, thanks to Lemma 24, for each player $i$ and each history $hv$, we can define for the subgame $\mathcal{G}_{\upharpoonright h}$ the notion of value denoted $val_i^h(v)$ and of optimal strategies denoted $\sigma_i^{hv}$, $\sigma_{-i}^{hv}$ for player $i$ and $-i$ respectively. We also denote by $\nu_i^{hv}$ the equivalence class for $\sim_i$, equal to the projection of $val_i^h(v)$. This means that $\sigma_i^{hv}$, (resp. $\sigma_{-i}^{hv}$) guarantees consistent plays $\pi$ starting at $v$ such that

$$val_i^h(v) \precsim_i^h [\pi]_i^h \quad \text{(resp. } [\pi]_i^h \precsim_i^h val_i^h(v)),$$

or equivalently

$$\nu_i^{hv} \precsim_i [h\pi]_i \quad \text{(resp. } [h\pi]_i \precsim_i \nu_i^{hv}). \tag{5}$$

For each $i$, let us define the following strategy $\tau_i$[18] in $\mathcal{G}$: for all histories $hv$ with $v \in V_i$,

$$\tau_i(hv) = \sigma_i^{h_1u}(h_2v)$$

such that

- $h = h_1h_2$,
- $\nu_i^{h_1u} = \nu_i^{hv}$ with $u$ the first vertex of $h_2v$,
- $|h_1|$ is minimal with respect to the two previous items.

(We use the optimal strategy corresponding to the smallest prefix $h_1$ of $h$ such that the projections of both values $val_i^h(v)$ and $val_i^{h_1}(u)$ are the same.)

Consider the outcome $\rho = \langle \tau \rangle_{v_0}$ of the strategy profile $\tau = (\tau_i)_{i \in \mathcal{P}}$ from a given initial vertex $v_0$. For each $i$, we define $\nu_i^* = \max\{\nu_i^{hv} \mid hv \text{ prefix of } \rho\}$. Notice that $\nu_i^*$ is well defined as any $\nu_i^{hv}$ is an equivalence class of $\sim_i$ and $\Lambda_i$ is a finite total order. Let us prove that

$$\nu_i^* \precsim_i [\rho]_i \text{ for all } i \in \mathcal{P}. \tag{6}$$

For this purpose, let us consider the smallest prefix $hv$ of $\rho$ such that $\nu_i^{hv} = \nu_i^*$, and let us prove that for any $hgu$ prefix of $\rho$,

- $\tau_i(hgu) = \sigma_i^{hv}(gu)$ if $u \in V_i$,
- $\nu_i^{hgu} = \nu_i^*$.

(From $h$, $\tau_i$ plays as dictated by $\sigma_i^{hv}$ and the projection of the value $val_i^{hg}(u)$ remains equal to $\nu_i^*$.) The proof is by induction on $|g|$. By definition of $\tau_i$ and $hv$, this property trivially holds when $g = \varepsilon$ (in which case $u = v$). Let us assume that the property is true for $gu$ and let us prove that it remains true for $guu'$, with $u' \in V$ such that $hguu'$ is a prefix of $\rho$.

- Let us first observe that $guu'$ is consistent with $\sigma_i^{hv}$. This is verified for $gu$ by induction hypothesis. Moreover, again by induction hypothesis, if $u \in V_i$, then $\tau_i(hgu) = \sigma_i^{hv}(gu) = u'$.
- Let us then prove that $\nu_i^{hguu'} = \nu_i^*$. By definition of $\nu_i^*$, we have $\nu_i^{hguu'} \precsim_i \nu_i^*$. Assume by contradiction that $\nu_i^{hguu'} \prec_i \nu_i^*$. By (5), it follows that the optimal strategy $\sigma_{-i}^{hguu'}$ can impose $[\pi]_i \precsim_i \nu_i^{hguu'} \prec_i v_i^*$ on plays $\pi$ having $hguu'$ as prefix. This is in contradiction with the optimal strategy $\sigma_i^{hv}$, the consistency of $guu'$ with this strategy, and thus by (5) $\nu_i^* = \nu_i^{hv} \precsim_i [\pi]_i$.

[18] Note that the strategy profile composed of the $\tau_i$'s is not yet the required NE.

- Suppose that $u' \in V_i$, as $\nu_i^{hguu'} = \nu_i^*$, it follows by definition of $\tau_i$ that $\tau_i(hguu') = \sigma_i^{hv}(guu')$.

As a consequence, the property is proved and we get that $\rho = h\rho'$ with a play $\rho'$ consistent with $\sigma_i^{hv}$. By (5) again, it follows that $\nu_i^* \precsim_i [\rho]_i$, and (6) is thus established.

We are now ready to define the required NE $\tau' = (\tau_i')_{i \in \mathcal{P}}$. It is partially defined in such a way as to generate $\rho = \langle \tau' \rangle_{v_0}$. Moreover, as soon as some player $i$ deviates from $\rho$ at any prefix $hv$ of $\rho$, then the other players form the coalition $-i$ and play the optimal strategy $\sigma_{-i}^{hv}$. In this way, by (5), the coalition imposes $[\rho']_i \precsim_i \nu_i^{hv}$ on the deviating play $\rho'$. As $\nu_i^{hv} \precsim_i \nu_i^*$ by definition of $\nu_i^*$ and by (6), we get $[\rho']_i \precsim_i \nu_i^{hv} \precsim_i \nu_i^* \precsim_i [\rho]_i$. The deviating play is therefore not profitable for player $i$, showing that $\tau'$ is an NE from $v_0$. ◀

## I Proofs of Proposition 12 and Theorem 11

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

**Proof.** The equivalence classes of $\sim$ form a finite lattice that we can see as an acyclic graph (whose vertices are the equivalence classes and $(C, C')$ is an edge if and only if $C \prec C'$). Thus, by performing a topological sort of this graph, we can totally order the equivalence classes as the sequence $C_1, \dots, C_n$, such that $C_i \precsim C_j$ implies $i \leq j$ (the topological sort respects $\precsim$). Let $i \leq n$, and $\pi \in C_i$ be a lasso (by Lemma 17). Therefore, we construct $\precsim'$ as follows:

$$\precsim' = \precsim \cup \left( \bigcup_{1 \leq i < j \leq n} C_i \times C_j \right).$$

We clearly get an embedding of $\precsim$ into $\precsim'$ such that $\precsim'$ is $\omega$-recognizable (as each equivalence class $C_i$ is $\omega$-regular by $\omega$-recognizably of $\precsim$). The relation $\precsim'$ is also clearly a preorder as $\precsim$ is a preorder and by the topological sort. Let us note that $\precsim'$ is total by construction. Indeed, for any $x, y$, there exist $C_i$ and $C_j$ such that $x \in C_i$ and $y \in C_j$. Therefore, w.l.o.g., assuming $i \leq j$, we have $x \precsim' y$ because either $(x, y) \in C_i \times C_j$ if $i \neq j$, or $x \sim y$ thus $x \precsim' y$ if $i = j$.

The second claim is clearly verified by construction for $\bowtie$ in $\{\precsim, \succsim, \sim\}$, as $\precsim'$ preserves $\precsim$. For $\prec$ and $\succ$, it is true by the same argument and the topological sort preserving the partial order of the equivalence classes. ◀

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

For the proof, we suppose that the preference relations $\precsim_i$, $i \in \mathcal{P}$, are not necessarily total. This means that when $x \not\prec_i y$, then either $x \not\precsim_i y$ or $y \precsim_i x$.

**Proof.** Given a game $\mathcal{G} = (A, (\precsim_i)_{i \in \mathcal{P}})$, we construct $\mathcal{G}' = (A, (\precsim_i')_{i \in \mathcal{P}})$ with the same arena $A$ and, for each $i \in \mathcal{P}$, the $\omega$-recognizable total preference relations $\precsim_i'$ as defined in Proposition 12. By Theorem 18, as $\precsim_i'$ are total, there exists an NE $\sigma = (\sigma_i)_{i \in \mathcal{P}}$ in $\mathcal{G}'$. This strategy profile $\sigma$ is also an NE in $\mathcal{G}$. Otherwise, we would get a profitable deviation $\langle \sigma \rangle_{v_0} \prec_i \pi$ for some player $i$ in $\mathcal{G}$, thus the same profitable deviation $\langle \sigma \rangle_{v_0} \prec_i' \pi$ in $\mathcal{G}'$ by Proposition 12. This is a contradiction with $\sigma$ being an NE in $\mathcal{G}'$. As there exists an NE, there also exists an NE composed of finite-memory strategies by Corollary 8. ◀

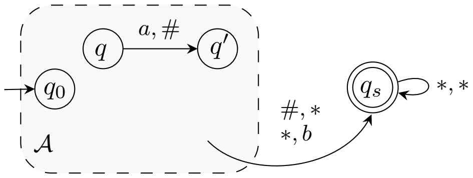

**Figure 14** The DBA $\mathcal{A}'$ used in the reduction for the prefix-independency property.

## J Characterization of NE Outcomes and Prefix-Independency

We begin with a proof of Theorem 13, then we will discuss about the alternative statement with prefix-linear relations.

▶ **Theorem 13.** *Let $\mathcal{G}$ be a game such that each preference relation $\precsim_i$ is an $\omega$-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 $\rho_n$ of $\rho$, if $\rho_n \in V_i$, then $val_i(\rho_n) \precsim_i [\rho]_i$.*

**Proof.** Suppose first that $\rho$ is the outcome of an NE $(\tau_i)_{i\in\mathcal{P}}$. Assume by contradiction that for some vertex $v = \rho_n \in V_i$ belonging to $\rho$, we have $val_i(v) \not\precsim_i [\rho]_i$, i.e., $[\rho]_i \prec_i val_i(v)$ as $\precsim_i$ is total. Let $\rho = h\rho_{\geq n}$, and let us consider an optimal strategy $\sigma_i^v$ for player $i$. Player $i$ can use this strategy to deviate at $v$ and thus produce a play $\pi$ starting at $v$ such that $val_i(v) \precsim_i [\pi]_i$. As $\precsim_i$ is prefix-independent, we have $\pi \sim_i h\pi$, and thus $[\rho]_i \prec_i val_i(v) \precsim_i [h\pi]_i$. Hence $h\pi$ is a profitable deviation for $i$, which is impossible since $\rho$ is an NE outcome.

Suppose now that $\rho$ is a play such that for all vertices $\rho_n$ of $\rho$, if $\rho_n \in V_i$, then $val_i(\rho_n) \precsim_i [\rho]_i$. We define a strategy profile $\tau = (\tau_i)_{i\in\mathcal{P}}$ such that it produces $\rho$, and if player $i$ deviates at some vertex $v$ from $\rho$, then the coalition $-i$ plays an optimal strategy $\sigma_{-i}^v$ from $v$. Let us prove that $\tau$ is an NE. Suppose that player $i$ deviates at $v = \rho_n$ from $\rho$ and consider the deviating play $\rho' = h\pi'$. By definition of $\tau$, we have $[\pi']_i \precsim_i val_i(\rho_n)$. As $\precsim_i$ is prefix-independent, $[\rho']_i = [h\pi']_i \precsim_i val_i(\rho_n)$. By hypothesis, $val_i(\rho_n) \precsim_i \rho$. Therefore, $[\rho']_i \precsim_i [\rho]_i$, showing that $\rho'$ is not a profitable deviation. Hence, $\rho$ is an NE outcome. ◀

As explained in Section 7, we can replace the assumption of prefix-independency with prefix-linearity and slightly modify the statement of Theorem 13. This leads to the next theorem, whose proof is almost the same.

▶ **Theorem 25.** *Let $\mathcal{G}$ be a game such that each preference relation $\precsim_i$ is an $\omega$-recognizable preorder, total, and prefix-linear. Then a play $\rho = \rho_0\rho_1\dots$ is an NE outcome if and only if for all vertices $\rho_n$ of $\rho$, if $\rho_n \in V_i$, then $val_i(\rho_n) \precsim_i [\rho_{\geq n}]_i$.*

In view of Theorems 13 and 25, we might want to check whether a given $\omega$-automatic relation satisfies the prefix-independency or prefix-linearity property, as for the classical properties considered in Proposition 9. We get the same complexity class:

▶ **Proposition 26.** *The problem of deciding whether an $\omega$-automatic relation $R$ is prefix-independent (resp. prefix-linear) is* NL*-complete.*

**Proof of Proposition 26 - prefix-independency.** We begin with the NL-membership. Let $\mathcal{A}$ be a DPA accepting $R$. Note that $R$ is prefix-independent if and only if we have the following equivalence for all $x, y \in \Sigma^\omega$:

$$(x, y) \in R \;\Leftrightarrow\; (x_{\geq 1}, y) \in R \wedge (x, y_{\geq 1}) \in R.$$

We are going to show that it is decidable whether $R$ is not prefix-independent, i.e., the following language $L$ is not empty:

$$\begin{aligned} L = \{(x, y, x_{\geq 1}, y_{\geq 1}) \in (\Sigma^\omega)^4 \mid \; & \Big((x,y) \in R \wedge ((x_{\geq 1}, y) \notin R \vee (x, y_{\geq 1}) \notin R)\Big) \\ & \vee \Big((x_{\geq 1}, y) \in R \wedge (x, y_{\geq 1}) \in R \wedge (x,y) \notin R\Big) \quad \}. \end{aligned} \tag{7}$$

As $R$ is accepted by a DPA, the set $\{(x,y) \mid (x,y) \notin R\}$ is accepted by a DPA of the same size as $\mathcal{A}$. Therefore, one can construct a generalized DPA $\mathcal{A}'$ accepting $L$ with a Boolean combination of six parity conditions (by carefully constructing the transitions to deal with tuples $(x, y, x_{\geq 1}, y_{\geq 1})$). Hence, testing whether $L$ is not empty can be done in $\mathsf{NL}$ by Proposition 16 while constructing $\mathcal{A}'$ on the fly.

Let us now show the $\mathsf{NL}$-hardness, by reduction from the universality problem of DBAs [36]. Let $\mathcal{A}$ be a DBA over the alphabet $\Sigma$. Let $\Sigma' = \Sigma \cup \{\#\}$ with a new symbol $\#$. From $R' = (\Sigma^\omega \backslash \mathcal{L}(\mathcal{A})) \times \{\#^\omega\}$, we define the complementary relation

$$R = ((\Sigma')^\omega \times (\Sigma')^\omega) \setminus R'.$$

Let us show that the reduction is correct. If $\mathcal{L}(\mathcal{A}) = \Sigma^\omega$, then $R = (\Sigma')^\omega \times (\Sigma')^\omega$, which is prefix-independent. Conversely, if there exists $x \in \Sigma^\omega \backslash \mathcal{L}(\mathcal{A})$, then $(x, \#^\omega) \notin R$ but $(x, a\#^\omega) \in R$ with $a \in \Sigma$, so $R$ is not prefix-independent. Let us now describe a DBA $\mathcal{A}'$ accepting $R$, as depicted in Figure 14. W.l.o.g., we suppose that $\mathcal{A}$ is complete. The automaton $\mathcal{A}'$ is composed of a copy of $\mathcal{A}$ extended with a second component $\#$ in a way to accept $\mathcal{L}(\mathcal{A}) \times \{\#^\omega\}$. As soon as a pair $(\#, *)$ or $(*, b)$ (with $b \in \Sigma$) is read from this copy, we go in an accepting sink state $q_s$. Therefore, the only rejected words are those in $R'$. Note that this is a logspace reduction as we need a logarithmic space on the work tape to construct $\mathcal{A}'$. ◀

**Proof of Proposition 26 - prefix-linearity.** Let us begin with the $\mathsf{NL}$-membership. Let $R$ be a preference relation and $\mathcal{A}$ be a DPA accepting it. Let us show that $R$ is not prefix-linear, i.e., there exist $x, y \in \Sigma^\omega$ and $u \in \Sigma^*$ such that $(x,y) \in R$ and $(ux, uy) \notin R$. We first compute the set $I = \{q \mid q$ is reachable by a run on $(u,u)$, for $u \in \Sigma^\omega\}$ by deleting all edges labeled $(a,b)$ for $a \neq b$. Then, we construct a nondeterministic parity automaton $\mathcal{B}$ by taking the complement of $\mathcal{A}$ that accepts $\neg R$ and saying that every state $q \in I$ is initial. Therefore, $(x,y) \in \mathcal{L}(\mathcal{B})$ if and only if there exists $u \in \Sigma^*$ such that $(ux, uy) \notin R$. Thus, we have to check whether $\mathcal{L}(\mathcal{A} \cap \mathcal{B}) \neq \varnothing$. This is the nonemptiness problem of a generalized NPA with a conjunction of two parity objectives, an $\mathsf{NL}$-complete problem by Proposition 16.

Let us now prove the $\mathsf{NL}$-hardness with a reduction from the emptiness of a DBA $\mathcal{A}$, an $\mathsf{NL}$-complete problem [36]. We construct $R = \mathcal{L}(\mathcal{A}) \times \{\#^\omega\}$. Clearly, $R$ is accepted by a DBA (thus a DPA) consisting in a copy of $\mathcal{A}$ where any label $a$ is replaced by $(a, \#)$ for all $a \in \Sigma$. Moreover, if there exists a word $w \in \mathcal{L}(\mathcal{A})$, then $(w, \#^\omega) \in R$ but $(\#w, \#^\omega) \notin R$, so $R$ is not prefix-linear. Conversely, if $\mathcal{L}(\mathcal{A}) = \varnothing$, then $R = \varnothing$ and thus it is prefix-linear. ◀