Pareto-Rational Verification

Véronique Bruyère
Université de Mons (UMONS), Mons, Belgium

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

Clément Tamines
Université de Mons (UMONS), Mons, Belgium

Abstract

We study the rational verification problem which consists in verifying the correctness of a system executing in an environment that is assumed to behave rationally. We consider the model of rationality in which the environment only executes behaviors that are Pareto-optimal with regard to its set of objectives, given the behavior of the system (which is committed in advance of any interaction). We examine two ways of specifying this behavior, first by means of a deterministic Moore machine, and then by lifting its determinism. In the latter case the machine may embed several different behaviors for the system, and the universal rational verification problem aims at verifying that all of them are correct when the environment is rational. For parity objectives, we prove that the Pareto-rational verification problem is co-NP-complete and that its universal version is in PSPACE and both NP-hard and co-NP-hard. For Boolean Büchi objectives, the former problem is Π_2P-complete and the latter is PSPACE-complete. We also study the case where the objectives are expressed using LTL formulas and show that the first problem is PSPACE-complete, and that the second is 2EXPTIME-complete. Both problems are also shown to be fixed-parameter tractable for parity and Boolean Büchi objectives.

2012 ACM Subject Classification Software and its engineering → Formal methods; Theory of computation → Logic and verification; Theory of computation → Solution concepts in game theory

Keywords and phrases Rational verification, Model-checking, Pareto-optimality, ω-regular objectives

Digital Object Identifier 10.4230/LIPIcs.CONCUR.2022.33

Related Version *Full Version*: `https://arxiv.org/abs/2202.13485` [19]

Funding This work is partially supported by the two PDR projects *Subgame perfection in graph games* and *Rational* (F.R.S.-FNRS), the ARC project *Non-Zero Sum Game Graphs: Applications to Reactive Synthesis and Beyond* (Fédération Wallonie-Bruxelles), the EOS project *Verifying Learning Artificial Intelligence Systems* (F.R.S.-FNRS and FWO), and the COST Action 16228 *GAMENET* (European Cooperation in Science and Technology).

1 Introduction

Formal verification is essential to ensure the correctness of systems responsible for critical tasks. Many advancements have been made in the field of formal verification both in terms of theoretical foundations and tool development, and computer-aided verification techniques, such as *model-checking* [4, 7], are now widely used in industry. In the classical approach to verification, it is assumed that the system designer provides *(i)* a model of the system to verify, together with *(ii)* a model of the environment in which the system will be executed, and *(iii)* a specification φ (e.g. an ω-regular property) that must be enforced by the system. Those models are usually nondeterministic automata that cover all possible behaviors of both the system and the environment. The model-checking algorithm is then used to decide if all executions of the system in the environment are correct with regard to φ. Unfortunately, in some settings, providing a faithful and sufficiently precise model of the environment may be *difficult or even impossible*. This is particularly true in heterogeneous systems composed of software entities interacting with human users, e.g. self-driving cars interacting with human drivers. Alternative approaches are thus needed in order to verify such complex multi-agent systems. One possible solution to this problem is to consider more *declarative* ways of modeling the environment. Instead of considering an operational model of each agent composing the environment, in this paper, we propose instead to identify the *objectives* of those agents. We then consider only the behaviors of the environment that concur to those objectives, instead of all behaviors described by some model of the system. We study the problem of *rational verification*: the system needs to be proven correct with regard to property φ, not in all the executions of the environment, but only in those executions that are *rational* with regard to the objectives of the environment.

There are several ways to model rationality. For instance, a famous model of rational behavior for the agents is the concept of *Nash equilibrium* (NE) [39]. Some promising exploratory works, based on the concept of NE, exist in the literature, like in verification of non-repudiation and fair exchange protocols [35, 23], planning of self-driving cars interacting with human drivers [45], or the automatic verification of an LTL specification in multi-agent systems that behave according to an NE [32]. Another classical approach is to model the environment as a single agent with multiple objectives. In that setting, trade-offs between (partially) conflicting objectives need to be made, and a rational agent will behave in a way to satisfy a *Pareto-optimal* set of its objectives. Pareto-optimality and multi-objective formalisms have been considered in computer science, see for instance [41] and references therein, and in formal methods, see e.g. [2, 12].

Nevertheless, we have only scratched the surface and there is a lack of a general theoretical background for marrying concepts from game theory and formal verification. This is the motivation of our work. We consider the setting in which a designer specifies the behavior of a system and identifies its objective Ω_0 as well as the multiple objectives $\{\Omega_1, \dots, \Omega_t\}$ of the environment in a underlying game arena G. The behavior of the system is usually modeled by the designer using a *deterministic Moore machine* that describes the strategy of the system opposite the environment. The designer can also use the model of *nondeterministic* Moore machine in order to describe a set of multiple possible strategies for the system instead of some single specific strategy. Given a strategy σ_0 for the system, the environment being rational only executes behaviors induced by σ_0 which result in a Pareto-optimal payoff with regard to its set of objectives $\{\Omega_1, \dots, \Omega_t\}$. When the Moore machine $\mathcal{M}$ is deterministic, the *Pareto-rational verification* (PRV) problem asks whether all behaviors that are induced by the machine $\mathcal{M}$ in G and that are Pareto-optimal for the environment all satisfy the objective Ω_0 of the system (a toy example giving intuition on this problem is proposed in the full version). When the Moore machine is nondeterministic, the *universal* PRV problem asks whether for all strategies σ_0 of the system described by $\mathcal{M}$, all behaviors induced by σ_0 that are Pareto-optimal for the environment satisfy Ω_0. The latter problem is a clear generalization of the former and is conceptually more challenging, as it asks to verify the correctness of the possibly infinite set of strategies described by $\mathcal{M}$. The universal PRV problem is also a well motivated problem, as typically, in the early stages of a development cycle, not all implementation details are fixed, and the use of nondeterminism is prevailing. In this last setting, we want to guarantee that a positive verification result is transferred to all possible implementations of the nondeterministic model of the system.

Technical Contributions. We introduce the Pareto-rational verification (PRV) problem and its universal variant. The objective Ω_0 of the system and the set $\{\Omega_1, \dots, \Omega_t\}$ of objectives of the environment are ω-regular objectives. We consider several ways of specifying these

Table 1 Summary of complexity results for the PRV problem and UPRV problem.

Objective	PRV problem complexity	UPRV problem complexity
Parity	co-NP-complete (Theorem 5)	PSPACE, NP-hard, co-NP-hard (Theorem 10)
Boolean Büchi	Π_2P-complete (Theorem 5)	PSPACE-complete (Theorem 10)
LTL	PSPACE-complete (Theorem 15)	2EXPTIME-complete (Theorem 14)

objectives: either by using parity conditions (a canonical way to specify ω-regular objectives), Boolean Büchi conditions (a generic way to specify Büchi, co-Büchi, Streett, Rabin, and other objectives), or using LTL formulas. Our technical results, some of which are summarized in Table 1, are as follows.

First, we study the complexity class of the PRV problem. We prove that it is co-NP-complete for parity objectives, Π_2P-complete for Boolean Büchi objectives, and PSPACE-complete for LTL objectives.

Second, we consider the universal variant of the PRV problem. We prove that it is in PSPACE and both NP-hard and co-NP-hard for parity objectives, PSPACE-complete for Boolean Büchi objectives, and 2EXPTIME-complete for LTL objectives.

Third, we establish the fixed-parameter tractability (FPT) of the universal PRV problem where the parameters are the number t of objectives of the environment as well as the highest priorities used in the parity objectives or the size of the formulas used in the Boolean Büchi objectives. For the particular case of the PRV problem with parity conditions, the parameters reduce to t only. Since this number is expected to be limited in practice, our result is of practical relevance. We additionally provide an alternative, possibly more efficient in practice, FPT algorithm for solving the PRV problem which exploits counterexamples and builds an under-approximation of the set of Pareto-optimal payoffs on demand.

Related Work. The concept of nondeterminism for strategies has been studied in the particular context of two-player zero-sum games where one player is opposed to the other one, under the name of permissive strategy, multi-strategy, or nondeterministic strategy in [5, 9, 10, 38, 44]. Those works concern synthesis and not verification.

Several fundamental results have been obtained on multi-player games played on graphs where the objectives of the players are Boolean or quantitative (see e.g. the book chapter [31] or the surveys [11, 13, 14]). Several notions of rational behavior of the players have been studied such as NEs, subgame perfect equilibria (SPEs) [46], secure equilibria [21], or profiles of admissible strategies [6]. The existing results in the literature are mainly focused on the existence of equilibria or the synthesis of such equilibria when they exist. Multidimensional energy and mean-payoff objectives for two-player games played on graphs have been studied in [20, 49, 50] and the Pareto curve of multidimensional mean-payoff games has been studied in [12]. Two-player games with heterogeneous multidimensional quantitative objectives have been investigated in [16].

Recent results concern the synthesis of strategies for a system in a way to satisfy its objective when facing an environment that is assumed to behave rationally with respect to the objectives of all his components. In [29, 36, 37], the objectives are expressed as LTL formulas and the considered models of rationality are NEs or SPEs. Algorithmic questions about this approach are studied in [24] for different types of ω-regular objectives. In [18], the objectives are ω-regular and the environment is assumed to behave rationally by playing in a way to obtain Pareto-optimal payoffs with respect to its objectives. We consider the concepts of [18] as a foundation for Pareto-rational verification.

The previously mentioned results all deal with the existence or the synthesis of solutions. Rational verification (instead of synthesis) is studied in [32] (see also the survey [1]), where the authors study how to verify a given specification for a multi-agent system with agents that behave rationally according to an NE when all objectives are specified by LTL formulas. They prove that this problem is 2EXPTIME-complete and design an algorithm that reduces this problem to solving a collection of parity games. This approach is implemented in the Equilibrium Verification Environment tool. In this paper, we study Pareto-optimality as a model of rationality instead of the concepts of NE or SPE. Our framework is more tractable as the PRV problem is PSPACE-complete for LTL specifications.

2 Definitions and the Pareto-Rational Verification Problem

We start by recalling several classical concepts of game theory, and in particular the model of (nondeterministic) Moore machines. We then present the verification problem studied in this paper and illustrate it on an example. We end the section by discussing the complexity of useful checks performed in several algorithms throughout this paper.

2.1 Definitions

Game Arena and Plays. A *game arena* is a tuple $G = (V, V_0, V_1, E, v_0)$ where (V, E) is a finite directed graph such that: *(i)* V is the set of vertices and (V_0, V_1) forms a partition of V where V_0 (resp. V_1) is the set of vertices controlled by Player 0 (resp. Player 1), *(ii)* $E \subseteq V \times V$ is the set of edges such that each vertex v has at least one successor v', i.e., $(v, v') \in E$, and *(iii)* $v_0 \in V$ is the initial vertex. We denote by $|G|$ the size of G. A *sub-arena* G' with a set $V' \subseteq V$ of vertices and initial vertex $v'_0 \in V'$ is a game arena defined from G as expected. A *single-player* game arena is a game arena where $V_0 = \emptyset$ and $V_1 = V$.

A *play* in a game arena G is an infinite sequence of vertices $\rho = v_0 v_1 \cdots \in V^\omega$ such that it starts with the initial vertex v_0 and $(v_j, v_{j+1}) \in E$ for all $j \in \mathbb{N}$. *Histories* in G are finite non-empty sequences $h = v_0 \dots v_j \in V^+$ defined similarly. The set of plays in G is denoted by Plays_G and the set of histories (resp. histories ending with a vertex in V_i) is denoted by Hist_G (resp. $\mathsf{Hist}_{G,i}$). Notations Plays, Hist, and Hist_i are used when G is clear from the context. The set of vertices occurring (resp. occurring infinitely often) in a play ρ is written $\mathsf{Occ}(\rho)$ (resp. $\mathsf{Inf}(\rho)$).

Strategies and Moore Machines. A *strategy* σ_i for Player i is a function $\sigma_i \colon \mathsf{Hist}_i \to V$ assigning to each history $hv \in \mathsf{Hist}_i$ a vertex $v' = \sigma_i(hv)$ such that $(v, v') \in E$. A play $\rho = v_0 v_1 \dots$ is *consistent* with σ_i if $v_{j+1} = \sigma_i(v_0 \dots v_j)$ for all $j \in \mathbb{N}$ such that $v_j \in V_i$. Consistency is naturally extended to histories. The set of plays (resp. histories) consistent with strategy σ_i is written $\mathsf{Plays}_{\sigma_i}$ (resp. Hist_{σ_i}).

A strategy σ_i for Player i is *finite-memory* [30] if it can be encoded by a *deterministic Moore machine* $\mathcal{M} = (M, m_0, \alpha_U, \alpha_N)$ where M is the finite set of states (the memory of the strategy), $m_0 \in M$ is the initial memory state, $\alpha_U : M \times V \to M$ is the update function, and $\alpha_N : M \times V_i \to V$ is the next-move function. Such a machine defines the strategy σ_i such that $\sigma_i(hv) = \alpha_N(\widehat{\alpha}_U(m_0, h), v)$ for all histories $hv \in \mathsf{Hist}_i$, where $\widehat{\alpha}_U$ extends α_U to histories as expected. In this paper, we consider the broader notion of *nondeterministic* Moore machine $\mathcal{M}$ (see e.g. [5]) with a next-move function $\alpha_N : M \times V_i \to 2^V$. Such a machine embeds a (possibly infinite) set of strategies σ_i for Player i such that $\sigma_i(hv) \in \alpha_N(\widehat{\alpha}_U(m_0, h), v)$ for all histories $hv \in \mathsf{Hist}_i$[1]. We denote by $[\![\mathcal{M}]\!]$ the set of all strategies defined by $\mathcal{M}$. The *size* of $\mathcal{M}$ is equal to the number $|M|$ of its memory states. Example 1 illustrates these concepts.

[1] Notice that this definition is different from simply making the machine deterministic by fixing a single next vertex $v' \in \alpha_N(m, v)$ for each $m \in M$ and $v \in V_i$.

When $\mathcal{M}$ is a deterministic Moore machine with $|M| = 1$, then it defines a *memoryless* strategy σ_i where $\sigma_i(hv) = \sigma_i(h'v)$ for all $hv, h'v$ ending with the same vertex $v \in V_i$. When $\mathcal{M}$ is a nondeterministic Moore machine with $|M| = 1$ and such that $\alpha_N(m_0, v) = \{v' \mid (v, v') \in E\}$, then $[\![\mathcal{M}]\!]$ is exactly the set of *all possible strategies* for Player i.

Objectives. An *objective* for Player i is a set of plays $\Omega \subseteq \mathsf{Plays}$. A play ρ *satisfies* the objective Ω if $\rho \in \Omega$. The *opposite objective* of Ω is written $\overline{\Omega} = \mathsf{Plays} \setminus \Omega$. We consider the following objectives in this paper:

- Let $c : V \to \{0, \dots, d\}$ be a function called a *priority function* which assigns an integer to each vertex in the arena (we assume that d is even). The set of priorities occurring infinitely often in a play ρ is $\mathsf{Inf}(c(\rho)) = \{c(v) \mid v \in \mathsf{Inf}(\rho)\}$. The *parity* objective $\mathsf{Parity}(c) = \{\rho \in \mathsf{Plays} \mid \min(\mathsf{Inf}(c(\rho))) \text{ is even}\}$ asks that the minimum priority visited infinitely often be even. The opposite objective $\overline{\Omega}$ of a parity objective Ω is again a parity objective (the priority function c' of $\overline{\Omega}$ is such that $c'(v) = c(v) + 1$ for all $v \in V$).
- Given m sets $T_1, \dots, T_m$ such that $T_i \subseteq V$, $i \in \{1, \dots, m\}$, and ϕ a Boolean formula over the set of variables $X = \{x_1, \dots, x_m\}$, the *Boolean Büchi*[2] [27, 17] objective $\mathsf{BooleanBüchi}(\phi, T_1, \dots, T_m) = \{\rho \in \mathsf{Plays} \mid \rho \text{ satisfies } (\phi, T_1, \dots, T_m)\}$ is the set of plays whose valuation of the variables in X satisfy formula ϕ. Given a play ρ, its valuation is such that $x_i = 1$ if and only if $\mathsf{Inf}(\rho) \cap T_i \neq \emptyset$ and $x_i = 0$ otherwise. That is, a play satisfies the objective if the Boolean formula describing the sets to be visited (in)finitely often by a play is satisfied. It is assumed that negations only appear in literals of ϕ and we denote by $|\phi|$ the size of ϕ equal to the number of symbols in $\{\wedge, \vee, \neg\} \cup X$ in ϕ.
  The opposite objective $\overline{\Omega}$ of a Boolean Büchi objective Ω is again a Boolean Büchi objective (the formula $\neg\phi$ of $\overline{\Omega}$ is obtained from ϕ by replacing each symbol $\vee$ (resp. $\wedge$) by $\wedge$ (resp. $\vee$) and each literal by its negation).

We recall that parity and Boolean Büchi objectives Ω are *prefix-independent*, i.e., whenever $\rho \in \Omega$, then all suffixes of ρ also satisfy Ω.

Zero-Sum Games. A two-player *zero-sum game* $\mathcal{G} = (G, \Omega)$ is a game on a game arena G where Player 0 has objective Ω and Player 1 has the opposite objective $\overline{\Omega}$. Given an initial vertex v_0, we say that a player is *winning from* v_0 if he has a strategy such that all plays starting with v_0 and consistent with this strategy satisfy his objective. We assume that the reader is familiar with this concept, see e.g. [30].

Lattices and Antichains. A *complete lattice* is a partially ordered set $(S, \leq)$ where S is a set, $\leq\, \subseteq S \times S$ is a partial order on S, and for every pair of elements $s, s' \in S$, their greatest lower bound and their least upper bound both exist. A subset $A \subseteq S$ is an *antichain* if all of its elements are pairwise incomparable with respect to $\leq$. Given $T \subseteq S$ and an antichain $A \subseteq S$, we denote by $\lceil T \rceil$ the set of maximal elements of T (which is thus an antichain) and by $\downarrow^{<} A$ the set of all elements $s \in S$ for which there exists some $s' \in A$ such that $s < s'$. Given two antichains $A, A' \subseteq S$, we write $A \sqsubseteq A'$ when for all $s \in A$, there exists $s' \in A'$ such that $s \leq s'$, and we write $A \sqsubset A'$ when $A \sqsubseteq A'$ and $A \neq A'$.

2.2 Pareto-Rational Verification Problem

We start by recalling the class of two-player games considered in this paper and the notion of payoffs in those games.

[2] This objective is also called *Emerson-Lei* objective.

Stackelberg-Pareto Games. A *Stackelberg-Pareto game* (SP game) $\mathcal{G} = (G, \Omega_0, \Omega_1, \dots, \Omega_t)$ is composed of a game arena G, an objective Ω_0 for Player 0, and $t \geq 1$ objectives $\Omega_1, \dots, \Omega_t$ for Player 1 [18]. An SP game where all objectives are parity (resp. Boolean Büchi) objectives is called a *parity* (resp. *Boolean Büchi*) *SP game*.

Payoffs. The *payoff* of a play $\rho \in \mathsf{Plays}$ is the vector of Booleans $\mathsf{pay}(\rho) \in \{0,1\}^t$ such that for all $i \in \{1, \dots, t\}$, $\mathsf{pay}_i(\rho) = 1$ if $\rho \in \Omega_i$, and $\mathsf{pay}_i(\rho) = 0$ otherwise. Notice that we omit to include the objective of Player 0 when discussing the payoff of a play. Instead we say that a play ρ is *won* by Player 0 if $\rho \in \Omega_0$ and we write $\mathsf{won}(\rho) = 1$, otherwise it is *lost* by Player 0 and we write $\mathsf{won}(\rho) = 0$. We write $(\mathsf{won}(\rho), \mathsf{pay}(\rho))$ for the *extended payoff* of ρ. A payoff p (resp. extended payoff (w, p)) is *realizable* if there exists a play $\rho \in \mathsf{Plays}$ such that $\mathsf{pay}(\rho) = p$ (resp. $(\mathsf{won}(\rho), \mathsf{pay}(\rho)) = (w, p)$); we say that ρ *realizes* p (resp. (w, p)).

We consider the following partial order on payoffs. Given two payoffs $p = (p_1, \dots, p_t)$ and $p' = (p'_1, \dots, p'_t)$ such that $p, p' \in \{0,1\}^t$, we say that p' is *larger* than p and write $p \leq p'$ if $p_i \leq p'_i$ for all $i \in \{1, \dots, t\}$. Moreover, when it also holds that $p_i < p'_i$ for some i, we say that p' is *strictly larger* than p and we write $p < p'$. Notice that the pair $(\{0,1\}^t, \leq)$ is a complete lattice with size 2^t and that the size of any antichain on $(\{0,1\}^t, \leq)$ is thus upper bounded by 2^t.

Let $\mathcal{G} = (G, \Omega_0, \Omega_1, \dots, \Omega_t)$ be an SP game and let σ_0 be a strategy of Player 0. We can consider the set of payoffs of plays consistent with σ_0 which are *Pareto-optimal*, i.e., maximal with respect to $\leq$. We write this set $P_{\sigma_0} = \max\{\mathsf{pay}(\rho) \mid \rho \in \mathsf{Plays}_{\sigma_0}\}$. Notice that this set is an antichain. In this paper, we study the following *verification* problem.

▶ **Problem.** Let $\mathcal{G}$ be an SP game and let $\mathcal{M}$ be a nondeterministic Moore machine for Player 0. The *universal Pareto-rational verification problem* (UPRV problem) is to decide whether for all $\sigma_0 \in [\![\mathcal{M}]\!]$, it holds that every play $\rho \in \mathsf{Plays}_{\sigma_0}$ such that $\mathsf{pay}(\rho) \in P_{\sigma_0}$ satisfies the objective of Player 0. When $\mathcal{M}$ is deterministic, we consider the single strategy $\sigma_0 \in [\![\mathcal{M}]\!]$ and speak about the *Pareto-rational verification problem* (PRV problem).

The UPRV problem models the situation where the system may employ one of several possible strategies in a nondeterministic manner and we therefore want to verify that all of them are correct. We do so in the context where the environment is rational and only executes behaviors which result in a Pareto-optimal payoff with regard to its set of objectives. In the following sections, we study the complexity of the (U)PRV problem in terms of $|G|$ the size of the game arena, $|M|$ the size of the Moore machine, t the number of objectives of Player 1, $\max d_i$ the maximum of all maximum priorities d_i according to each parity objective Ω_i in case of parity SP games, and $\max |\phi_i|$ the maximum of all sizes $|\phi_i|$ such that ϕ_i is the formula for objective Ω_i in case of Boolean Büchi SP games.

▶ **Example 1.** Consider the parity SP game $\mathcal{G}$ with arena G depicted in Figure 1 (left) in which Player 1 has $t = 3$ objectives [18]. The vertices of Player 0 (resp. Player 1) are depicted as circles (resp. squares)[3]. We do not explicitly define the parity objective Ω_0 of Player 0 nor the three parity objectives of Player 1. Instead, the extended payoff of plays reaching vertices from which they can only loop is displayed in the arena next to those vertices, and we set the extended payoff of play $v_0v_2(v_3v_5)^\omega$ to $(0,(0,1,0))$.

[3] This convention is used throughout this paper.

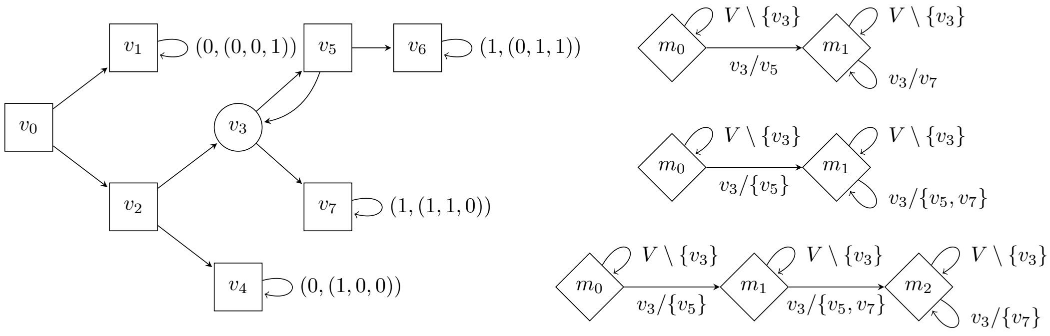

Figure 1 A parity SP game (left), one deterministic Moore machine $\mathcal{M}_t$ and two nondeterministic Moore machines $\mathcal{M}_c$ and $\mathcal{M}_b$ (respectively top, center, and bottom right).

Consider the memoryless strategy σ_0 of Player 0 such that he chooses to always move to v_5 from v_3. The set of payoffs of plays consistent with σ_0 is $\{(0,0,1),(0,1,0),(1,0,0),(0,1,1)\}$ and the set of those that are Pareto-optimal is $P_{\sigma_0} = \{(1,0,0),(0,1,1)\}$. Notice that play $\rho = v_0v_2v_4^\omega$ is consistent with σ_0, has payoff $(1,0,0) \in P_{\sigma_0}$ and is lost by Player 0. Together with $\mathcal{G}$, strategy σ_0 is therefore a negative instance of the PRV problem.

Let us now consider the finite-memory strategy σ_0' such that $\sigma_0'(v_0v_2v_3) = v_5$ and $\sigma_0'(v_0v_2v_3v_5v_3) = v_7$. Contrarily to the previous strategy, $\mathcal{G}$ and σ_0' constitute a positive instance of the PRV problem. Indeed, the set of Pareto-optimal payoffs is $P_{\sigma_0'} = \{(0,1,1),(1,1,0)\}$ and Player 0 wins every play consistent with σ_0' whose payoff is in this set. A deterministic Moore machine $\mathcal{M}_t$ for σ_0' is depicted in Figure 1 (top right). It has two memory states with state m_1 indicating that v_3 has been visited. Each edge from m to m' is labeled by v/v' with an optional v' such that $\alpha_U(m,v) = m'$ and $\alpha_N(m,v) = v'$ if $v \in V_0$.

Finally, we provide two nondeterministic Moore machines in Figure 1 (center right and bottom right). Each edge from m to m' is now labeled by v/T such that $\alpha_N(m,v) = T \subseteq V$ when $v \in V_0$. Let us show that the SP game $\mathcal{G}$ with machine $\mathcal{M}_c$ (resp. machine $\mathcal{M}_b$) is a negative (resp. positive) instance of the UPRV problem.

One can check that the memoryless strategy σ_0 mentioned above (always move to v_5 from v_3) belongs to the set $[\![\mathcal{M}_c]\!]$. It follows that $\mathcal{G}$ and $\mathcal{M}_c$ are a negative instance of the UPRV problem. Notice that all the other strategies σ_0^k, $k \geq 1$, of $[\![\mathcal{M}_c]\!]$ are such that $\sigma_0^k(hv_3) = v_5$ except when $h = v_0v_2(v_3v_5)^k$ in which case $\sigma_0^k(hv_3) = v_7$ (the strategy allows to cycle between v_3 and v_5 k times before dictating that v_7 be visited).

The machine $\mathcal{M}_b$ has three memory states such that m_1 (resp. m_2) records one visit (resp. at least two visits) to v_3. The set $[\![\mathcal{M}_b]\!]$ contains exactly two strategies: one is the finite-memory strategy σ_0' given before and the other one is the strategy σ_0'' such that $\sigma_0''(v_0v_2v_3) = \sigma_0''(v_0v_2v_3v_5v_3) = v_5$ and $\sigma_0''(v_0v_2v_3(v_5v_3)^2) = v_7$. One can verify that $\mathcal{G}$ and $\mathcal{M}_b$ are a positive instance of the UPRV problem. ⌟

▶ Remark 2. In the sequel, we often consider the Cartesian product $G \times \mathcal{M}$ with initial vertex (v_0, m_0) of the arena G of $\mathcal{G}$ with the (nondeterministic) Moore machine $\mathcal{M}$ for Player 0. When $\mathcal{M}$ is nondeterministic, this finite graph $G \times \mathcal{M}$ is a two-player game arena (as the vertices of Player 0 may have several successors). The strategies σ_0' for Player 0 in this product correspond exactly to the strategies $\sigma_0 \in [\![\mathcal{M}]\!]$. With this in mind, we can reformulate the UPRV problem to take a game arena as input. Given $G' = G \times \mathcal{M}$, the UPRV problem is to decide whether for all strategies σ_0' of Player 0 in G', every play $\rho \in \mathsf{Plays}_{\sigma_0'}$ such that $\mathsf{pay}(\rho) \in P_{\sigma_0'}$ satisfies the objective of Player 0. When $\mathcal{M}$ is deterministic, this product is a finite graph whose infinite paths, starting from the initial vertex, are exactly the plays consistent with the single strategy $\sigma_0 \in [\![\mathcal{M}]\!]$. This graph can be seen as a single-player game arena G' (as every vertex of Player 0 only has a single successor). In that setting, given a single-player arena $G' = G \times \mathcal{M}$, the PRV problem is to decide whether every play $\rho \in \mathsf{Plays}_{G'}$ such that $\mathsf{pay}(\rho) \in \max\{\mathsf{pay}(\rho) \mid \rho \in \mathsf{Plays}_{G'}\}$ satisfies the objective of Player 0.

Payoff Realizability and Lassoes. In order to study the (U)PRV problem, we need to perform specific checks on payoffs as described in the next proposition (the proof of which can be found in the full version).

▶ **Proposition 3.** *Let $\mathcal{G} = (G, \Omega_1, \dots, \Omega_t)$ be an SP game and let p (resp. (w, p)) be a payoff (resp. extended payoff). The existence of a play ρ realizing payoff p (resp. extended payoff (w, p)) can be decided with the following complexities.*

- *For parity objectives: in time polynomial in $|G|$, t, and $\max d_i$.*
- *For Boolean Büchi objectives: in time polynomial in $|G|$, and exponential in t and $\max |\phi_i|$.*

Checking whether a realizable payoff p is Pareto-optimal is decided with the same complexities.

We also need the next property which shows that when a play satisfies a parity or a Boolean Büchi objective, there exists another such play that is a lasso of polynomial size.

▶ **Lemma 4** ([8]). *For any play $\rho \in \mathsf{Plays}$, there exists a lasso $\rho' = gh^\omega$ such that ρ and ρ' start with the same vertex, $\mathsf{Occ}(\rho) = \mathsf{Occ}(\rho')$, $\mathsf{Inf}(\rho) = \mathsf{Inf}(\rho')$, and $|gh|$ is quadratic in $|G|$.*

Related Synthesis Problem. Our verification problem is related to the *Stackelberg-Pareto Synthesis problem* introduced in [18]. This synthesis problem asks, given a two-player SP game, whether there exists a strategy σ_0 for Player 0 such that every play in $\mathsf{Plays}_{\sigma_0}$ with a Pareto-optimal payoff satisfies the objective of Player 0. This problem is solved in [18] for parity and reachability objectives. It is shown that the problem is NEXPTIME-complete, and that finite-memory strategies are sufficient for Player 0 to have a solution σ_0 to the problem.

3 Complexity Class of the PRV problem

In this section, we provide the complexity class of the PRV problem for both parity SP games and Boolean Büchi SP games. The complexity class of the UPRV problem is studied in Section 4. In this whole section, we assume that an instance of the PRV problem is an SP game with a *single-player* game arena (see Remark 2). This is not problematic with respect to the algorithmic complexities since the size of the single-player game arena is $|G| \cdot |M|$.

▶ **Theorem 5.** *The PRV problem is* co-NP*-complete for parity SP games and* Π_2P*-complete for Boolean Büchi SP games.*

We now detail the arguments used to show the co-NP-completeness for parity SP games, and refer the reader to the full version for the completeness result for Boolean Büchi SP games.

Membership to co-NP. The co-NP-membership stated in Theorem 5 is easily proved by showing that the complement of the PRV problem is in NP. Given a single-player SP game $\mathcal{G}$, we guess a payoff $p \in \{0, 1\}^t$, and we check *(i)* whether p is realizable and Pareto-optimal, and *(ii)* whether there exists a play ρ with payoff p which is lost by Player 0. In the case of parity objectives, those two checks can be performed in polynomial time by Proposition 3.

The proof of co-NP-hardness is more involved. In order to show this result, we provide a reduction from the co-3SAT problem to the PRV problem.

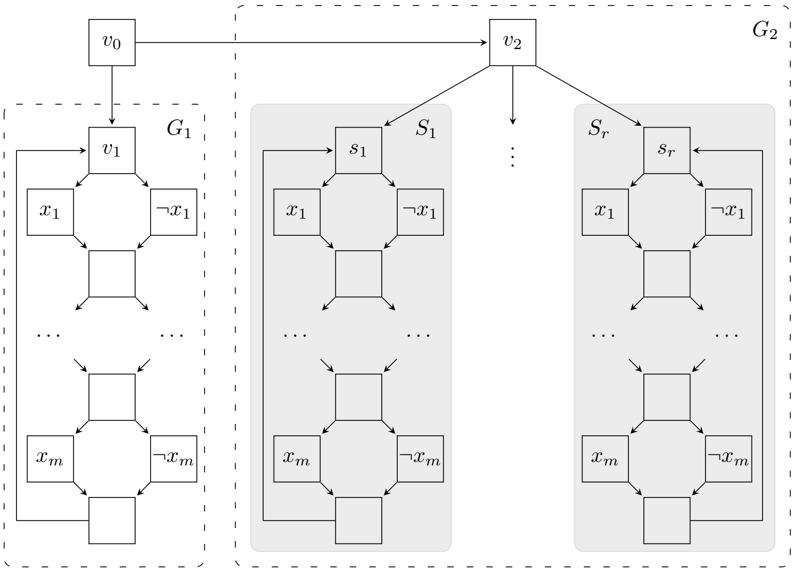

Figure 2 The single-player arena G used in the reduction from co-3SAT for parity objectives.

The co-3SAT Problem. We consider a formula $\psi = D_1 \wedge \cdots \wedge D_r$ in 3-Conjunctive Normal Form (3CNF) consisting of r clauses, each containing exactly 3 literals over the set of variables $X = \{x_1, \ldots, x_m\}$. We assume that each variable x occurs as a literal $\ell \in \{x, \neg x\}$ in at least one clause of ψ. The satisfiability problem, called 3SAT, is to decide whether there exists a valuation of the variables in X such that the formula ψ evaluates to true. This problem is well-known to be NP-complete [25, 34]. We can consider the complement of this problem, which is to decide for such a formula ψ whether *all valuations* of the variables in X falsify the formula i.e., make at least one of the clauses evaluate to false. This problem, called co-3SAT, being the complement of an NP-complete problem, is co-NP-complete [40].

Intuition of the Reduction. Given an instance of co-3SAT, we create a parity SP game $\mathcal{G}$ with a single-player game arena G consisting of two sub-arenas G_1 and G_2 reachable from the initial vertex v_0 as depicted in Figure 2. The intuition behind this construction is the following. A play in the arena starts in v_0 and will either enter G_1 through v_1 and stay in that sub-arena forever or enter G_2 through v_2, visit some vertex s_i with $i \in \{1, \ldots, r\}$, and stay forever in the corresponding sub-arena S_i. The objectives are devised such that a payoff contains one objective per literal of X and one objective per literal, per clause of ψ. A play in G_1 has a payoff corresponding to a valuation of X and the literals in the clauses of ψ satisfied by that valuation. In addition, the objective of Player 0 is not satisfied in those plays. Therefore, it must be the case that the payoffs of plays in G_1 are not Pareto-optimal in order for the instance of the PRV problem to be positive. This is only the case when the instance of co-3SAT is also positive due to the fact that plays in G_2, which all satisfy the objective of Player 0, then have payoffs strictly larger than that of plays in G_1. This is not the case if some play in G_1 corresponds to a valuation of X which satisfies ψ.

Structure of a Payoff. We now detail the objectives used in the reduction and the corresponding structure of a payoff in G. Player 0 has a single parity objective Ω_0. Player 1 has $1 + 2 \cdot m + 3 \cdot r$ parity objectives (assuming each clause is composed of exactly 3 literals). The payoff of a play in G therefore consists in a vector of $1 + 2 \cdot m + 3 \cdot r$ Booleans for the following objectives:

$$(\Omega_1, \Omega_{x_1}, \Omega_{\neg x_1}, \ldots, \Omega_{x_m}, \Omega_{\neg x_m}, \Omega_{\ell^{1,1}}, \Omega_{\ell^{1,2}}, \Omega_{\ell^{1,3}}, \ldots, \Omega_{\ell^{r,1}}, \Omega_{\ell^{r,2}}, \Omega_{\ell^{r,3}}).$$

The objective Ω_0 is equal to objective $\Omega_1 = \mathsf{Parity}(c)$ with $c(v) = 2$ if $v \in G_2$ and $c(v) = 1$ otherwise. It is direct to see that these objectives are only satisfied for plays in G_2. We define the objective $\Omega_x = \mathsf{Parity}(c)$ (resp. $\Omega_{\neg x} = \mathsf{Parity}(c')$) with $c(x) = 2$ and $c(\neg x) = 1$ (resp. $c'(\neg x) = 2$ and $c'(x) = 1$) for the vertices labelled x and $\neg x$ in G_1 and G_2, and such that every other vertex has priority 2 according to c (resp. c'). Objective Ω_x (resp. $\neg\Omega_x$) is satisfied if and only if vertex x (resp. $\neg x$) is visited infinitely often and $\neg x$ (resp. x) is not. If both x and $\neg x$ are visited infinitely often, neither Ω_x not $\Omega_{\neg x}$ are satisfied. These objectives are used to encode valuations of X into payoffs. The objective $\Omega_{\ell^{i,j}}$ corresponds to the objective for the j^{th} literal of the i^{th} clause of ψ, written $\ell^{i,j} \in \{x_k, \neg x_k\}$ for some $k \in \{1, \ldots, m\}$, we define the priority function for this objective later for each sub-arena.

Payoff of Plays Entering Sub-Arena G_1. We define the priority function c of objective $\Omega_{\ell^{i,j}}$ in G_1 such that $c(\ell^{i,j}) = 2$ and $c(\neg\ell^{i,j}) = 1$ for vertices labeled $\ell^{i,j}$ and $\neg\ell^{i,j}$ in G_1. Notice that a play in G_1 corresponds to repeatedly making the choice of visiting x_i or $\neg x_i$ for $i \in \{1, \ldots, m\}$. We call plays which visit both x_i and $\neg x_i$ infinitely often for some i *unstable* plays and those which visit infinitely often either x_i or $\neg x_i$ for each i *stable* plays. We introduce the following lemma on the stability of plays in G_1 (proved in the full version).

▶ **Lemma 6.** *Unstable plays in G_1 do not have a Pareto-optimal payoff.*

In the sequel, we therefore only consider stable plays ρ in G_1. The objective Ω_0 of Player 0 and Ω_1 of Player 1 are not satisfied in ρ and such a play satisfies either the objective Ω_{x_i} or $\Omega_{\neg x_i}$ for each $x_i \in X$. The part of the payoff of ρ for these objectives can be seen as a valuation of the variables in X, expressed as a vector of $2 \cdot m$ Booleans. The objective $\Omega_{\ell^{i,j}}$ is satisfied in the payoff of ρ if and only if the literal $\ell^{i,j}$ is satisfied by that valuation. That is if either $\ell^{i,j} = x_k$ and Ω_{x_k} is satisfied or $\ell^{i,j} = \neg x_k$ and $\Omega_{\neg x_k}$ is satisfied, for $x_k \in X$. Given a positive instance of the co-3SAT problem, it holds that none of the valuations of X satisfy the formula ψ. Therefore, since stable plays in G_1 encode valuations of X and the corresponding satisfied literals of the clauses of ψ, the next lemma holds (see full version).

▶ **Lemma 7.** *Given a positive instance of the co-3SAT problem and any stable play ρ in G_1, there exists a clause D_i for $i \in \{1, \ldots, r\}$ such that $\Omega_{\ell^{i,j}}$ is not satisfied in ρ for $j \in \{1, 2, 3\}$.*

In order for the instance of the PRV problem to be positive in case of a positive instance of co-3SAT, since plays in G_1 do not satisfy the objective of Player 0, it must be the case that the payoff of these plays are not Pareto-optimal when considering the whole arena G. Therefore, given any play in G_1, there must exists a play with a strictly larger payoff in G_2 which also satisfies the objective of Player 0.

Payoff of Plays Entering Sub-Arena G_2. We define the priority function c of objective $\Omega_{\ell^{i,j}}$ in G_2 such that $c(s_i) = 1$ and $c(v) = 2$ for $v \neq s_i$ in G_2. Therefore, any play entering S_i satisfies every objective for the literals of the clauses of ψ, except for objectives $\Omega_{\ell^{i,j}}$, $j \in \{1, 2, 3\}$. After entering a sub-arena S_j, plays in G_2 can visit infinitely often either or both x_i and $\neg x_i$ for $i \in \{1, \ldots, m\}$ and we therefore introduce the following lemma on the stability of plays in G_2, the proof of which is given in the full version.

▶ **Lemma 8.** *Unstable plays in G_2 do not have a Pareto-optimal payoff.*

We therefore only consider stable plays in G_2. Such a play ρ satisfies either the objective Ω_{x_i} or $\Omega_{\neg x_i}$ for each $x_i \in X$. The objectives corresponding to the literals in the clauses of ψ which are satisfied in ρ only depend on the sub-arena S_j entered by ρ. It can easily be shown that every such objective is satisfied by ρ except for $\Omega_{\ell j,1}, \Omega_{\ell j,2}$ and $\Omega_{\ell j,3}$ for clause D_j.

Correctness. Finally, we briefly discuss the correctness of this reduction (a full proof is provided in the full version). In case of a positive instance of the co-3SAT problem, for every valuation of X (and therefore every stable play ρ in G_1), this valuation does not satisfy some clause D_i of ϕ (and therefore ρ does not satisfy any objective $\Omega_{\ell i,j}$ by Lemma 7). It follows that there exists a play with a strictly larger payoff in G_2 given the form of the payoff of plays in G_2 discussed above (and the fact that they satisfy objective Ω_1 while ρ does not). In case of a negative instance of co-3SAT, this is not the case as some stable play in G_1 corresponds to a valuation which satisfies ϕ and therefore satisfies at least one objective for each clause D_i. As plays in G_2 do not satisfy any objective for some clause, G_1 contains a Pareto-optimal play lost by Player 0, and the instance of the PRV problem is negative.

▶ Remark 9. As stated in Theorem 5, the lower bound for the PRV problem is stronger for Boolean Büchi objectives than for parity objectives. We can show that this difference in complexity is even more apparent if we consider the following variant of the *complement* of the PRV problem in which we *fix a payoff* for Player 1: given a single-player SP game and a payoff p, decide whether there exists a play with payoff p not satisfying Ω_0 and p is Pareto-optimal. While this problem is in P for parity SP games (indeed p does not need to be guessed anymore), it is BH_2-complete for Boolean Büchi SP games. We refer the reader to the full version for details about this additional complexity result.

4 Complexity Class of the UPRV problem

We study in this section the complexity class of the UPRV problem for parity and Boolean Büchi SP games. Our results are summarized in the following theorem.

▶ **Theorem 10.** *The UPRV problem is*
- PSPACE*-complete for Boolean Büchi SP games,*
- *in* PSPACE*,* NP*-hard and* co-NP*-hard for parity SP games.*

We show the PSPACE-membership stated in Theorem 10 in the following proposition.

▶ **Proposition 11.** *The UPRV problem is in* PSPACE *for both Boolean Büchi SP games and parity SP games.*

Proof. Let $\mathcal{G}$ be an SP game and $\mathcal{M}$ be a nondeterministic Moore machine for Player 0. By Remark 2, the strategies of $[\![\mathcal{M}]\!]$ are exactly the strategies of the product $G' = G \times \mathcal{M}$. In the sequel, we will shift from G to G' and conversely without mentioning it explicitly.

To prove Proposition 11, it is enough to show that the complement of the UPRV problem is in NPSPACE, since NPSPACE = PSPACE and as the PSPACE class is closed under complementation. The complement of the UPRV problem is to decide whether there exists a strategy $\sigma_0 \in [\![\mathcal{M}]\!]$ and a play $\rho \in \mathsf{Plays}_{\sigma_0}$ such that $\mathsf{pay}(\rho) \in P_{\sigma_0}$ and ρ is lost by Player 0.

Our algorithm works as follows in G' (we detail its correctness and complexity later):
1. guess a lasso $\rho' = g'h'^{\omega}$ in $\mathsf{Plays}_{G'}$ such that $g'h'$ has polynomial size,
2. check that ρ' is lost by Player 0,

3. check that for each vertex v of ρ' controlled by Player 1, Player 0 is winning from v in the two-player *zero-sum* game $\mathcal{H} = (G', \Omega')$ with arena G' and objective $\Omega' = \{\rho^* \in \mathsf{Plays}_{G'} \mid \neg(\mathsf{pay}(\rho^*) > \mathsf{pay}(\rho'))\}$.

Let us prove that this algorithm is correct. *(i)* Assume first that there exists a strategy $\sigma_0 \in [\![\mathcal{M}]\!]$ and a play $\rho \in \mathsf{Plays}_{\sigma_0}$ such that $\mathsf{pay}(\rho) \in P_{\sigma_0}$ and ρ is lost by Player 0. We see this play ρ as a play in G'. By Lemma 4 there exists a lasso $\rho' = g'h'^{\omega}$ of polynomial size in G' which realises the same extended payoff and such that $\mathsf{Occ}(\rho) = \mathsf{Occ}(\rho')$. This lasso is what is guessed in step 1 of the algorithm. By our assumptions on ρ, we know that it satisfies the check of step 2. It remains to explain why the second check also succeeds in step 3. From each vertex v of ρ' (and thus of ρ) controlled by Player 1, Player 0 is winning in $\mathcal{H}$ thanks to his strategy σ_0. Indeed, any play $\rho'_1 \in \mathsf{Plays}_{G'}$ consistent with σ_0 cannot have a payoff strictly larger than $\mathsf{pay}(\rho') \in P_{\sigma_0}$, and parity and Boolean Büchi objectives are prefix-independent. *(ii)* Assume now that the two checks of our algorithm succeed for the guessed lasso ρ'. Let us define a strategy σ_0 for Player 0 in G' (which is also a strategy $\sigma_0 \in [\![\mathcal{M}]\!]$) as follows: first we define σ_0 in a way to produce play ρ'; second after each history hvv' such that hv is prefix of ρ' and hvv' is not (meaning that v belongs to Player 1), σ_0 acts as the winning strategy of Player 0 from v in $\mathcal{H}$. We have thus proved that there exist a strategy $\sigma_0 \in [\![\mathcal{M}]\!]$ and a play $\rho' \in \mathsf{Plays}_{\sigma_0}$ such that $\mathsf{pay}(\rho') \in P_{\sigma_0}$ and ρ' is lost by Player 0.

Let us now show that our nondeterministic algorithm executes in polynomial space. Step 1 requires polynomial space to store $g'h'$. The check of step 2 requires to verify that $\rho' \notin \Omega_0$ such that Ω_0 is either a parity or a Boolean Büchi objective. This can be done by looking at the cycle h' in polynomial space. Let us now study step 3. We are going to show that $\mathcal{H} = (G', \Omega')$ is a zero-sum game with a Boolean Büchi objective Ω', known to be solvable in PSPACE [33]. Let us denote by $p = (p_1, \ldots, p_t)$ the payoff of ρ'. The objective Ω' is equal to

$$\Big(\bigcap_{p_i=0} \overline{\Omega}_i\Big) \cup \Big(\bigcup_{\substack{p_i=1\\p_j=0}} (\overline{\Omega}_i \cap \Omega_j)\Big) \tag{1}$$

where the the first disjunct expresses plays with payoffs less than or equal to p and the second disjunct expresses plays with payoffs incomparable with p. Recall that any parity objective can be expressed as a Boolean Büchi objective using a formula of size $\mathcal{O}(d^2)$ where d is the highest priority in the parity objective (see e.g. [3]). Therefore, for both parity and Boolean Büchi SP games, the objective Ω' is a Boolean Büchi objective defined by a formula of polynomial size. ◀

We now turn to the hardness results stated in Theorem 10. The co-NP hardness of the UPRV problem for parity SP games is easily obtained from the co-NP hardness of the PRV problem (Theorem 5). We consider the other hardness results in the following proposition.

▶ **Proposition 12.** *The UPRV problem is* NP*-hard for parity SP games, and* PSPACE*-hard for Boolean Büchi SP games.*

We prove the NP-hardness for parity SP games and refer the reader to the full version for the PSPACE-hardness for Boolean Büchi SP games. For this purpose, we reduce the following co-NP-hard problem to an instance of the complement of the UPRV problem.

Generalized Parity Game. Let us consider a two-player zero-sum generalized parity game $(G, \Omega_a \wedge \Omega_b)$ where the objective of Player 0 is a conjunction $\Omega_a \wedge \Omega_b$ of two parity objectives. Deciding whether Player 0 has a winning strategy from a vertex v_0 in G is co-NP-hard [22].

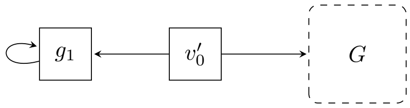

Figure 3 The arena G' used in the reduction from zero-sum games with two parity objectives.

Intuition of the Reduction. Given a zero-sum generalized parity game $(G, \Omega_a \wedge \Omega_b)$ and a vertex v_0, we construct an instance of the UPRV problem with the game arena G' depicted in Figure 3. In G', the dashed box labeled G represents the arena of the zero-sum game and we assume that the edge from v'_0 goes to v_0 in G. Equivalently, the dashed box is the Cartesian product of G and the nondeterministic machine $\mathcal{M}$ with one memory state embedding all possible strategies of Player 0 (see Remark 2). Notice that given a play ρ' of G' reaching G, we can retrieve a corresponding play ρ from v_0 in G. Any strategy σ_0 of Player 0 in G' is a strategy in $[\![\mathcal{M}]\!]$ and the converse also holds. We will see that the proposed construction is such that Player 0 has a winning strategy from v_0 in $(G, \Omega_a \wedge \Omega_b)$ if and only if the corresponding instance of the UPRV problem is *negative*.

Objectives. Player 0 has a single parity objective Ω_0 and Player 1 has two parity objectives Ω_1 and Ω_2. We first extend the priority function c_a of Ω_a (resp. c_b of Ω_b) to G' such that $c_a(g_1) = c_a(v'_0) = c_b(g_1) = c_b(v'_0) = 1$ and consider the corresponding objective Ω'_a (resp. Ω'_b) in G'. Notice that $\Omega'_a = \Omega_a$ (resp. $\Omega'_b = \Omega_b$) when considering only the plays of sub-arena G in G'. We define the actual objectives used in the reduction as follows. Player 0 has objective $\Omega_0 = \mathsf{Parity}(c)$ with a priority function c defined such that Ω_0 is only satisfied in plays reaching G. The first (resp. second) objective of Player 1 is such that $\Omega_1 = \overline{\Omega}'_a$ (resp. $\Omega_2 = \overline{\Omega}'_b$). Notice that objective Ω_1 (resp. Ω_2) is satisfied in plays reaching G if and only if the objective Ω_a (resp. Ω_b) is *not* satisfied in those plays. The play $v'_0 g_1^\omega$ is consistent with any strategy of Player 0 and has extended payoff $(0,(0,0))$. Any play reaching G is of the form $\rho' = v'_0\rho$ where ρ is a play in G starting from the initial vertex v_0. The extended payoff for such a play ρ' is $(1,(0,0))$ if ρ satisfies Ω_a and Ω_b; $(1,(0,1))$ if ρ satisfies Ω_a and not Ω_b; $(1,(1,0))$ if ρ satisfies Ω_b and not Ω_a; and $(1,(1,1))$ if ρ does not satisfy Ω_a nor Ω_b.

Correctness. If the instance of the UPRV problem is negative, it holds there exists a strategy $\sigma_0 \in [\![\mathcal{M}]\!]$ such that some play in $\mathsf{Plays}_{\sigma_0}$ has a Pareto-optimal payoff and is lost by Player 0. Since the play $v'_0 g_1^\omega$ with payoff $(0,0)$ is the only one in G' not to satisfy Ω_0, its payoff must be Pareto-optimal. It follows that all plays in G that are consistent with σ_0 have payoff $(0,0)$ and therefore satisfy the conjunction $\Omega_a \wedge \Omega_b$. Hence, σ_0 is a winning strategy for Player 0 from v_0 in the zero-sum game $(G, \Omega_a \wedge \Omega_b)$. Conversely, if Player 0 has a winning strategy from v_0 in $(G, \Omega_a \wedge \Omega_b)$, it holds that this strategy is in $[\![\mathcal{M}]\!]$ and such that all consistent plays in G satisfy the conjunction $\Omega_a \wedge \Omega_b$ and therefore has payoff $(0,0)$. It is easily checked that the instance of the UPRV problem is negative.

5 Fixed-Parameter Complexity

In this section, we study the fixed-parameter complexity of the (U)PRV problem. We refer the reader to [26] for the concept of fixed-parameter tractability (FPT). We recall that given an SP game $\mathcal{G} = (G, \Omega_0, \ldots, \Omega_t)$, $\max d_i$ is the maximum of all maximum priorities d_i according to each objective Ω_i in case of parity SP games, and that $\max |\phi_i|$ is the maximum of all sizes $|\phi_i|$ such that each ϕ_i defines objective Ω_i in case of Boolean Büchi SP games.

▶ **Theorem 13.** *The UPRV problem is in* FPT

- *with parameters t and $\max d_i$ for parity SP games (with an exponential in t and $\max d_i$),*
- *with parameters t and $\max |\phi_i|$ for Boolean Büchi SP games (with an exponential in t and $\max |\phi_i|$).*

The proof of this theorem uses a deterministic variant of the algorithm given in the proof of Proposition 11. Instead of guessing a lasso, we loop over each possible payoff p for which we test whether there exists a play ρ with payoff p not satisfying Ω_0, and such that Player 0 has a winning strategy from each vertex v of ρ in the zero-sum game $\mathcal{H} = (G \times \mathcal{M}, \Omega')$ with Ω' defined in (1). Whether Player 0 is winning from v in $\mathcal{H}$ can be checked with an FPT algorithm with parameter $|\phi'|$ (with an exponential in $|\phi'|$) where ϕ' defines the Boolean Büchi objective Ω' [17, 15].[4] Details of the proof are given in the full version.

A direct corollary of Theorem 13 is that the PRV problem is also in FPT. Nevertheless, we provide in the full version a simpler FPT algorithm for the PRV problem leading to an improved complexity for parity SP games (with a sole exponential in t). A second, more clever, variation is given in Algorithm 1 where instead of computing the antichain P_{σ_0} by going through the entire lattice of payoffs, we compute an *under-approximation* (with respect to $\sqsubseteq$) of P_{σ_0} on demand by using *counterexamples*. The algorithm systematically searches for plays ρ losing for Player 0 and maintains an antichain A of realizable payoffs to eliminate previous counterexamples. Initially, this antichain A is empty. A potential counterexample is a play ρ losing for Player 0 and such that for all payoffs p of A, $\mathsf{pay}(\rho)$ is not strictly smaller than p, that is, $\mathsf{pay}(\rho) \not\in {\downarrow^{<}} A$ (line 3). When a potential counterexample ρ exists, there are two possible cases. First, there exists a play ρ' winning for Player 0 and such that $\mathsf{pay}(\rho') > \mathsf{pay}(\rho)$ (line 4). The payoff of ρ' is added to A and a new approximation A of P_{σ_0} is computed (by keeping only the maximal elements, line 5). Second, if such a play ρ' does not exist, then we have identified a counterexample (the play ρ), showing that the instance of the PRV problem is negative (line 7). If there are no more potential counterexamples, then the instance is positive (line 9), otherwise we iterate. This algorithm is guaranteed to terminate as $A \sqsubset \lceil A \cup \{\mathsf{pay}(\rho')\} \rceil$ in line 5. Algorithm 1 is shown to be correct and in FPT in the full version of the paper, where we also evaluate it to show its efficiency in practice.

Algorithm 1 Counterexample-based algorithm for the PRV problem.

Input: A single-player SP game resulting from the Cartesian product of the arena G of an SP game and a deterministic Moore machine $\mathcal{M}$ for Player 0.
Output: Whether the instance of the PRV problem is positive.

1 $A \leftarrow \emptyset$
2 **repeat**
3   **if** $\exists \rho \in \mathsf{Plays}$ *such that* $\mathsf{won}(\rho) = 0$ *and* $\mathsf{pay}(\rho) \not\in {\downarrow^{<}} A$ **then**
4   **if** $\exists \rho' \in \mathsf{Plays}$ *such that* $\mathsf{won}(\rho') = 1$ *and* $\mathsf{pay}(\rho') > \mathsf{pay}(\rho)$ **then**
5             $A \leftarrow \lceil A \cup \{\mathsf{pay}(\rho')\} \rceil$
6   **else**
7   **return** False
8   **else**
9   **return** True

[4] The FPT algorithm in [17] is linear in the number of symbols $\vee, \wedge$ of ϕ' and double exponential in the number of variables of ϕ'. This complexity is improved in [15] by replacing the double exponential in $|\phi'|$ by a single one.

6 LTL Pareto-Rational Verification

We now show that when the objectives are expressed using Linear Temporal Logic (LTL) formulas, the PRV problem retains the PSPACE-completeness of the LTL model-checking problem, and the UPRV problem retains the 2EXPTIME-completeness of solving LTL games. We do not investigate the fixed-parameter complexity in this context as the completeness to PSPACE (resp. 2EXPTIME) already holds when Player 1 has a single objective.

LTL (Universal) Pareto-Rational Verification Problem. A *labeled game arena* G_λ is a game arena where a labeling function $\lambda : V \to 2^{AP}$ maps each vertex to a set of propositional variables in AP. An *LTL SP game* $\mathcal{G} = (G_\lambda, \phi_0, \phi_1, \ldots, \phi_t)$ is composed of a labeled game arena G_λ, an LTL formula ϕ_0 for Player 0 and $t \geq 1$ LTL formulas $\phi_1, \ldots, \phi_t$ for Player 1. The difference with regular SP games is thus that the goal of the players is expressed using LTL formulas over the set of propositional variables AP. The payoff of plays in G_λ is defined as expected. Given an LTL SP game, we consider the two verification problems described in Section 2 and call them the *LTL PRV problem* and *LTL UPRV problem*.

▶ **Theorem 14.** *The LTL UPRV problem is* 2EXPTIME*-complete.*

Proof. We first prove that the LTL UPRV problem is in 2EXPTIME. Given an LTL SP game $\mathcal{G}$ and a nondeterministic Moore machine $\mathcal{M}$, we proceed as follows. We first perform the Cartesian product $G' = G_\lambda \times \mathcal{A}_0 \times \mathcal{A}_1 \times \cdots \times \mathcal{A}_t$ of the arena G_λ with a Deterministic Parity Automaton (DPA) $\mathcal{A}_i$ for each LTL formula ϕ_i, $i \in \{0, \ldots, t\}$. The size of each automaton is at most double exponential in the size of its corresponding LTL formula, and the number of priorities it uses is exponential [48, 42, 28]. We thus have a parity SP game $\mathcal{G}'$ with arena G' of double exponential size. We then use the FPT algorithm of Theorem 13 on this SP game $\mathcal{G}'$, which is polynomial in $|G'|$ and exponential in the parameters t and $\max d'_i$ (the maximum priority used in the parity objectives). Therefore this algorithm is polynomial in $|G_\lambda|$, single exponential in t, and double exponential in the size of LTL formulas ϕ_i, $i \in \{0, \ldots, t\}$. This shows the 2EXPTIME-easyness.

Let us now prove the 2EXPTIME-hardness result by adapting the reduction of Proposition 12 for the case of the LTL UPRV problem.

- We consider the problem of deciding whether Player 0 has a winning strategy from v_0 in a two-player zero-sum game (G_λ, ϕ) where the ϕ is the LTL objective of Player 0. This problem is 2EXPTIME-complete [43].
- Given such a zero-sum game (G_λ, ϕ) and a vertex v_0, we construct an instance of the UPRV problem on the same game arena G' depicted in Figure 3. In this arena, G is replaced by G_λ and both v'_0 and g_1 are labelled with the set $\{x\}$ containing the single atomic proposition x which does not appear in ϕ. The nondeterministic machine $\mathcal{M}$ considered in the reduction is again the one with a single memory state that embeds every possible strategy of Player 0. The objective Ω_0 of Player 0 is defined by LTL formula ϕ_0 and the single objective Ω_1 of Player 1 is defined by LTL formula ϕ_1 as follows:
  - $\phi_0 = \neg \bigcirc x$,
  - $\phi_1 = (\neg \bigcirc x) \wedge (\neg \bigcirc \phi)$

  where $\bigcirc$ is the next operator in LTL. It is direct to see that objective Ω_0 is not satisfied by the play $v'_0 g_1^\omega$ and is satisfied by all plays reaching G_λ. The objective Ω_1 is not satisfied by the play $v'_0 g_1^\omega$ and is satisfied by plays reaching G_λ if and only if the formula ϕ is not satisfied in those plays.

- Using similar arguments as used in the proof of Proposition 12, the following holds. A strategy $\sigma_0 \in [\![\mathcal{M}]\!]$ makes the instance of the LTL UPRV problem negative if every play $v_0'\rho$ reaching G_λ and consistent with this strategy falsifies objective Ω_1 of Player 1 (as no payoff is then strictly larger than that of play $v_0' g_1^\omega$, lost by Player 0). If this is the case, it follows that strategy σ_0 is a winning strategy for Player 0 from v_0 in the zero-sum game (G_λ, ϕ) as every play ρ consistent with this strategy satisfies formula ϕ. The converse is also true. Player 0 therefore has a winning strategy from v_0 in (G_λ, ϕ) if and only if the corresponding instance of the LTL UPRV problem is negative. It follows that the LTL UPRV problem is 2EXPTIME-hard for LTL SP games (as co-2EXPTIME = 2EXPTIME). ◀

▶ **Theorem 15.** *The LTL PRV problem is* PSPACE*-complete.*

The proof of this theorem relies on two variants of the LTL model-checking problem that are both PSPACE-complete [47].

LTL Model-Checking Problem. Given a finite transition system T, an initial state, and an LTL formula ψ, the *LTL existential* (resp. *universal*) *model-checking problem* is to decide whether ψ is satisfied in at least one infinite path (resp. all infinite paths) of T starting from the initial state. Notice that a finite transition system is the same model as a single-player labeled game arena and that an infinite path in T corresponds to a play in this arena.

Proof of Theorem 15. We first show that the LTL PRV problem is in PSPACE. Given an LTL SP game $\mathcal{G}$, we proceed as follows. For each payoff $p \in \{0,1\}^t$, we check *(i)* whether it is realizable and Pareto-optimal, if yes *(ii)* whether there exists a play ρ such that $\mathsf{pay}(\rho) = p$ and $\mathsf{won}(\rho) = 0$. If for some payoff p, both tests succeed, then the given instance $\mathcal{G}$ is negative, otherwise it is positive (this approach is similar to the simpler FPT algorithm for the PRV problem provided in the full version). Checking that a payoff p is realizable reduces to solving the LTL existential model-checking problem for the formula $\psi = (\bigwedge_{p_i=1} \phi_i) \wedge (\bigwedge_{p_i=0} \neg\phi_i)$, this test can be performed in polynomial space. The second check in *(i)* and the last check in *(ii)* are similarly executed in polynomial space. The LTL PRV problem is hence in PSPACE.

We now prove that the LTL PRV problem is PSPACE-hard by showing that we can transform any instance of the LTL universal model-checking problem into an instance of the LTL PRV problem such that the instance of the former is positive if and only if the corresponding instance of the latter is positive as well. Let T be transition system and ψ be an LTL formula. Given our previous remark, T can be seen as a single-player labeled arena G_λ for some labeling function λ. We create the following LTL SP game $\mathcal{G} = (G_\lambda, \psi, \phi_1)$ played on $G_\lambda = T$ where the objective of Player 0 is to satisfy the formula ψ and the sole objective of Player 1 is to satisfy the formula $\phi_1 = \mathit{true}$. It is direct to see that any play in G_λ satisfies the objective of Player 1 and therefore that every play in G_λ is Pareto-optimal. It follows that the given instance of the LTL PRV problem is positive if and only if every play in G_λ satisfies the formula ψ. This corresponds exactly to the LTL universal model-checking problem. ◀

References

1 Alessandro Abate, Julian Gutierrez, Lewis Hammond, Paul Harrenstein, Marta Kwiatkowska, Muhammad Najib, Giuseppe Perelli, Thomas Steeples, and Michael J. Wooldridge. Rational verification: game-theoretic verification of multi-agent systems. *Appl. Intell.*, 51(9):6569–6584, 2021. doi:10.1007/s10489-021-02658-y.

2 Rajeev Alur, Aldric Degorre, Oded Maler, and Gera Weiss. On omega-languages defined by mean-payoff conditions. In Luca de Alfaro, editor, *Foundations of Software Science and Computational Structures, 12th International Conference, FOSSACS 2009, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2009, York, UK, March 22-29, 2009. Proceedings*, volume 5504 of *Lecture Notes in Computer Science*, pages 333–347. Springer, 2009. doi:10.1007/978-3-642-00596-1_24.

3 Christel Baier, Frantisek Blahoudek, Alexandre Duret-Lutz, Joachim Klein, David Müller, and Jan Strejcek. Generic emptiness check for fun and profit. In Yu-Fang Chen, Chih-Hong Cheng, and Javier Esparza, editors, *Automated Technology for Verification and Analysis – 17th International Symposium, ATVA 2019, Taipei, Taiwan, October 28-31, 2019, Proceedings*, volume 11781 of *Lecture Notes in Computer Science*, pages 445–461. Springer, 2019. doi:10.1007/978-3-030-31784-3_26.

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

5 Julien Bernet, David Janin, and Igor Walukiewicz. Permissive strategies: from parity games to safety games. *RAIRO Theor. Informatics Appl.*, 36(3):261–275, 2002. doi:10.1051/ita:2002013.

6 Dietmar Berwanger. Admissibility in infinite games. In Wolfgang Thomas and Pascal Weil, editors, *STACS 2007, 24th Annual Symposium on Theoretical Aspects of Computer Science, Aachen, Germany, February 22-24, 2007, Proceedings*, volume 4393 of *Lecture Notes in Computer Science*, pages 188–199. Springer, 2007. doi:10.1007/978-3-540-70918-3_17.

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

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

9 Patricia Bouyer, Marie Duflot, Nicolas Markey, and Gabriel Renault. Measuring permissivity in finite games. In Mario Bravetti and Gianluigi Zavattaro, editors, *CONCUR 2009 – Concurrency Theory, 20th International Conference, CONCUR 2009, Bologna, Italy, September 1-4, 2009. Proceedings*, volume 5710 of *Lecture Notes in Computer Science*, pages 196–210. Springer, 2009. doi:10.1007/978-3-642-04081-8_14.

10 Patricia Bouyer, Erwin Fang, and Nicolas Markey. Permissive strategies in timed automata and games. *Electron. Commun. Eur. Assoc. Softw. Sci. Technol.*, 72, 2015. doi:10.14279/tuj.eceasst.72.1015.

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. doi:10.1007/978-3-319-30000-9_1.

12 Romain Brenguier and Jean-François Raskin. Pareto curves of multidimensional mean-payoff games. In Daniel Kroening and Corina S. Pasareanu, editors, *Computer Aided Verification – 27th International Conference, CAV 2015, San Francisco, CA, USA, July 18-24, 2015, Proceedings, Part II*, volume 9207 of *Lecture Notes in Computer Science*, pages 251–267. Springer, 2015. doi:10.1007/978-3-319-21668-3_15.

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

14 Véronique Bruyère. Synthesis of equilibria in infinite-duration games on graphs. *ACM SIGLOG News*, 8(2):4–29, 2021. doi:10.1145/3467001.3467003.

15 Véronique Bruyère, Baptiste Fievet, Jean-François Raskin, and Clément Tamines. Stackelberg-Pareto synthesis (extended version). *CoRR*, abs/2203.01285, 2022. doi:10.48550/arXiv.2203.01285.

16 Véronique Bruyère, Quentin Hautem, and Jean-François Raskin. On the complexity of heterogeneous multidimensional games. In Josée Desharnais and Radha Jagadeesan, editors, *27th International Conference on Concurrency Theory, CONCUR 2016, August 23-26, 2016, Québec City, Canada*, volume 59 of *LIPIcs*, pages 11:1–11:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPIcs.CONCUR.2016.11.

17 Véronique Bruyère, Quentin Hautem, and Jean-François Raskin. Parameterized complexity of games with monotonically ordered omega-regular objectives. In Sven Schewe and Lijun Zhang, editors, *29th International Conference on Concurrency Theory, CONCUR 2018, September 4-7, 2018, Beijing, China*, volume 118 of *LIPIcs*, pages 29:1–29:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.CONCUR.2018.29.

18 Véronique Bruyère, Jean-François Raskin, and Clément Tamines. Stackelberg-Pareto synthesis. In Serge Haddad and Daniele Varacca, editors, *32nd International Conference on Concurrency Theory, CONCUR 2021, August 24-27, 2021, Virtual Conference*, volume 203 of *LIPIcs*, pages 27:1–27:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.CONCUR.2021.27.

19 Véronique Bruyère, Jean-François Raskin, and Clément Tamines. Pareto-rational verification. *CoRR*, abs/2202.13485, 2022. arXiv:2202.13485.

20 Krishnendu Chatterjee, Laurent Doyen, Thomas A. Henzinger, and Jean-François Raskin. Generalized mean-payoff and energy games. 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 505–516. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2010. doi:10.4230/LIPIcs.FSTTCS.2010.505.

21 Krishnendu Chatterjee, Thomas A. Henzinger, and Marcin Jurdzinski. Games with secure equilibria. *Theor. Comput. Sci.*, 365(1-2):67–82, 2006. doi:10.1016/j.tcs.2006.07.032.

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

25 Stephen A. Cook. The complexity of theorem-proving procedures. In Michael A. Harrison, Ranan B. Banerji, and Jeffrey D. Ullman, editors, *Proceedings of the 3rd Annual ACM Symposium on Theory of Computing, May 3-5, 1971, Shaker Heights, Ohio, USA*, pages 151–158. ACM, 1971. doi:10.1145/800157.805047.

26 R.G. Downey and M.R. Fellows. *Parameterized Complexity*. Monographs in Computer Science. Springer New York, 2012. URL: `https://books.google.be/books?id=HyTjBwAAQBAJ`.

27 E. Allen Emerson and Chin-Laung Lei. Modalities for model checking: Branching time logic strikes back. *Sci. Comput. Program.*, 8(3):275–306, 1987. `doi:10.1016/0167-6423(87)90036-0`.

28 Javier Esparza, Jan Kretínský, Jean-François Raskin, and Salomon Sickert. From LTL and limit-deterministic Büchi automata to deterministic parity automata. In Axel Legay and Tiziana Margaria, editors, *Tools and Algorithms for the Construction and Analysis of Systems – 23rd International Conference, TACAS 2017, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2017, Uppsala, Sweden, April 22-29, 2017, Proceedings, Part I*, volume 10205 of *Lecture Notes in Computer Science*, pages 426–442, 2017. `doi:10.1007/978-3-662-54577-5_25`.

29 Dana Fisman, Orna Kupferman, and Yoad Lustig. Rational synthesis. In Javier Esparza and Rupak Majumdar, editors, *Tools and Algorithms for the Construction and Analysis of Systems, 16th International Conference, TACAS 2010, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2010, Paphos, Cyprus, March 20-28, 2010. Proceedings*, volume 6015 of *Lecture Notes in Computer Science*, pages 190–204. Springer, 2010. `doi:10.1007/978-3-642-12002-2_16`.

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*, pages 151–178. Amsterdam University Press, 2008. URL: `http://www.jstor.org/stable/j.ctt46mwfz.11`.

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

33 Paul Hunter and Anuj Dawar. Complexity bounds for regular games. In Joanna Jedrzejowicz and Andrzej Szepietowski, editors, *Mathematical Foundations of Computer Science 2005, 30th International Symposium, MFCS 2005, Gdansk, Poland, August 29 – September 2, 2005, Proceedings*, volume 3618 of *Lecture Notes in Computer Science*, pages 495–506. Springer, 2005. `doi:10.1007/11549345_43`.

34 Richard M. Karp. Reducibility among combinatorial problems. In Raymond E. Miller and James W. Thatcher, editors, *Proceedings of a symposium on the Complexity of Computer Computations, held March 20-22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, USA*, The IBM Research Symposia Series, pages 85–103. Plenum Press, New York, 1972. `doi:10.1007/978-1-4684-2001-2_9`.

35 Steve Kremer and Jean-François Raskin. A game-based verification of non-repudiation and fair exchange protocols. *J. Comput. Secur.*, 11(3):399–430, 2003. URL: `http://content.iospress.com/articles/journal-of-computer-security/jcs185`, `doi:10.3233/jcs-2003-11307`.

36 Orna Kupferman, Giuseppe Perelli, and Moshe Y. Vardi. Synthesis with rational environments. *Ann. Math. Artif. Intell.*, 78(1):3–20, 2016. `doi:10.1007/s10472-016-9508-8`.

37 Orna Kupferman and Noam Shenwald. The complexity of LTL rational synthesis. In Dana Fisman and Grigore Rosu, editors, *Tools and Algorithms for the Construction and Analysis of Systems – 28th International Conference, TACAS 2022, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2022, Munich, Germany, April 2-7, 2022, Proceedings, Part I*, volume 13243 of *Lecture Notes in Computer Science*, pages 25–45. Springer, 2022. `doi:10.1007/978-3-030-99524-9_2`.

38 Michael Luttenberger. Strategy iteration using non-deterministic strategies for solving parity games. *CoRR*, abs/0806.2923, 2008. `arXiv:0806.2923`.

39 John F. Nash. Equilibrium points in n-person games. In *PNAS*, volume 36, pages 48–49. National Academy of Sciences, 1950.

40 Christos H. Papadimitriou. *Computational complexity*. Addison-Wesley, 1994.

41 Christos H. Papadimitriou and Mihalis Yannakakis. On the approximability of trade-offs and optimal access of web sources. In *41st Annual Symposium on Foundations of Computer Science, FOCS 2000, 12-14 November 2000, Redondo Beach, California, USA*, pages 86–92. IEEE Computer Society, 2000.

42 Nir Piterman. From nondeterministic Büchi and Streett automata to deterministic parity automata. *Log. Methods Comput. Sci.*, 3(3), 2007. doi:10.2168/LMCS-3(3:5)2007.

43 Amir Pnueli and Roni Rosner. On the synthesis of a reactive module. In *Conference Record of the Sixteenth Annual ACM Symposium on Principles of Programming Languages, Austin, Texas, USA, January 11-13, 1989*, pages 179–190. ACM Press, 1989. doi:10.1145/75277.75293.

44 Stéphane Riedweg and Sophie Pinchinat. You can always compute maximally permissive controllers under partial observation when they exist. In *Proc. 2005 American Control Conference, Portland, Oregon, June 8-10 2005*, volume 4, pages 2287–2292, Portland, Oregon, June 2005.

45 Dorsa Sadigh, Shankar Sastry, Sanjit A. Seshia, and Anca D. Dragan. Planning for autonomous cars that leverage effects on human actions. In David Hsu, Nancy M. Amato, Spring Berman, and Sam Ade Jacobs, editors, *Robotics: Science and Systems XII, University of Michigan, Ann Arbor, Michigan, USA, June 18 – June 22, 2016*, 2016. doi:10.15607/RSS.2016.XII.029.

46 Reinhard Selten. Spieltheoretische Behandlung eines Oligopolmodells mit Nachfrageträgheit. *Zeitschrift für die gesamte Staatswissenschaft*, 121:301–324 and 667–689, 1965.

47 A. Prasad Sistla and Edmund M. Clarke. The complexity of propositional linear temporal logics. *J. ACM*, 32(3):733–749, 1985. doi:10.1145/3828.3837.

48 A. Prasad Sistla, Moshe Y. Vardi, and Pierre Wolper. The complementation problem for Büchi automata with appplications to temporal logic. *Theor. Comput. Sci.*, 49:217–237, 1987. doi:10.1016/0304-3975(87)90008-9.

49 Yaron Velner, Krishnendu Chatterjee, Laurent Doyen, Thomas A. Henzinger, Alexander Moshe Rabinovich, and Jean-François Raskin. The complexity of multi-mean-payoff and multi-energy games. *Inf. Comput.*, 241:177–196, 2015. doi:10.1016/j.ic.2015.03.001.

50 Yaron Velner and Alexander Rabinovich. Church synthesis problem for noisy input. In Martin Hofmann, editor, *Foundations of Software Science and Computational Structures – 14th International Conference, FOSSACS 2011, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2011, Saarbrücken, Germany, March 26-April 3, 2011. Proceedings*, volume 6604 of *Lecture Notes in Computer Science*, pages 275–289. Springer, 2011. doi:10.1007/978-3-642-19805-2_19.