

On the Complexity of SPEs in Parity Games

Léonard Brice ✉

Université Gustave Eiffel, Marne-la-Vallée, France
Université Libre de Bruxelles, Belgium

Jean-François Raskin ✉

Université Libre de Bruxelles, Belgium

Marie van den Bogaard ✉

LIGM, Univ. Gustave Eiffel, CNRS, F-77454 Marne-la-Vallée, France

Abstract

We study the complexity of problems related to subgame-perfect equilibria (SPEs) in infinite duration non zero-sum multiplayer games played on finite graphs with parity objectives. We present new complexity results that close gaps in the literature. Our techniques are based on a recent characterization of SPEs in prefix-independent games that is grounded on the notions of requirements and negotiation, and according to which the plays supported by SPEs are exactly the plays consistent with the requirement that is the least fixed point of the negotiation function. The new results are as follows. First, checking that a given requirement is a fixed point of the negotiation function is an **NP**-complete problem. Second, we show that the SPE constrained existence problem is **NP**-complete, this problem was previously known to be **ExpTime**-easy and **NP**-hard. Third, the SPE constrained existence problem is fixed-parameter tractable when the number of players and of colors are parameters. Fourth, deciding whether some requirement is the least fixed point of the negotiation function is complete for the second level of the Boolean hierarchy. Finally, the SPE-verification problem – that is, the problem of deciding whether there exists a play supported by a SPE that satisfies some LTL formula – is **PSpace**-complete, this problem was known to be **ExpTime**-easy and **PSpace**-hard.

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 Games on graphs, subgame-perfect equilibria, parity objectives

Digital Object Identifier 10.4230/LIPIcs.CSL.2022.10

Related Version *Full Version*: <https://arxiv.org/abs/2107.07458> [3]

Funding This work is partially supported by 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 & FWO), the COST Action 16228 GAMENET (European Cooperation in Science and Technology), and by the PDR project *Subgame perfection in graph games* (F.R.S.- FNRS).

Acknowledgements We wish to thank anonymous reviewers for their useful observations.

1 Introduction

Nash equilibrium (NE) is one of the central concepts from game theory to formalize the notion of rationality. It describes profiles of strategies in which no player has an incentive to change their strategy unilaterally. However, in sequential games, like games played on graphs, NEs are known to be plagued by *non-credible threats*: players can threaten other players in subgames with *non-rational actions* in order to force an equilibrium that avoids these subgames. To avoid non-credible threats, subgame-perfect equilibria are used instead. *Subgame-perfect equilibria* (SPEs) are NEs that are NEs in all subgames of the original game: the players must act rationally in all subgames even after a deviation by another player.



© Léonard Brice, Jean-François Raskin, and Marie van den Bogaard;
licensed under Creative Commons License CC-BY 4.0

30th EACSL Annual Conference on Computer Science Logic (CSL 2022).

Editors: Florin Manea and Alex Simpson; Article No. 10; pp. 10:1–10:17



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

In this paper, we study the complexity of decision problems related to SPEs in sequential games played on graphs with parity objectives. In such a game, each vertex of the game graph has one color per player, and each player wants the least color they see infinitely often along a play, which is an infinite path in the graph, to be even. Parity conditions, in games as well as in automata, are canonical ways to represent ω -regular constraints. It is known that SPEs always exist in parity games, as shown in [14]. Unfortunately, the precise complexity of the SPE constrained existence problem, i.e. the problem of deciding whether there exists an SPE that generates payoffs between two given thresholds, is left open in the literature: it is known to be **ExpTime**-easy and **NP**-hard. We prove here that it is in fact **NP**-complete, and we provide several other new complexity results on related problems of interest.

While previous attempts to solve this decision problem were based on alternating tree automata ([9]), we obtain the new tight complexity results starting from concepts that we have introduced recently in [2] to capture SPEs in mean-payoff games: the notions of *requirements* and *negotiation*. A requirement is a function λ that maps each vertex v of the game graph to a real value, that represents the lowest payoff that the player controlling v should accept when facing other rational players. A play $\rho = \rho_0\rho_1\dots$ is λ -consistent if for each vertex ρ_k , the player controlling ρ_k gets at least the payoff $\lambda(\rho_k)$ in ρ . In Boolean games, such as parity games, we naturally consider requirements whose values are either 0 or 1 (1 meaning that the player must achieve their objective, 0 that they may not).

The negotiation function maps a requirement λ to a requirement $\text{nego}(\lambda)$, which captures from any vertex v the maximal payoff that the corresponding player can ensure, against λ -rational players, that is, players who play in such a way that they obtain at least the payoff specified by λ . Clearly, if λ_0 maps each vertex to 0, then every play is λ_0 -consistent. Then, the requirement $\text{nego}(\lambda_0)$ maps each vertex v to its *antagonistic value*, i.e. the best payoff that the player controlling v can ensure against an adversarial coalition of the other players (as any behavior of the other players is λ_0 -rational). It is the case that the $\text{nego}(\lambda_0)$ -consistent plays are exactly the plays supported by NEs.

But then, the following natural question is: given v and λ , can the player who controls v improve his worst-case value, if only plays that are consistent with λ are proposed by the other players? Or equivalently, can this player enforce a better value when playing against players that are not willing to give away their own worst-case value? which is clearly a minimal goal for any rational adversary. So $\text{nego}(\lambda)(v)$ returns this value; and this reasoning can be iterated. In [2], it is shown that the least fixed point λ^* of the negotiation function is exactly characterizing the set of plays supported by SPEs, for all prefix independent payoff functions with steady negotiation (which is the case for parity objectives).

Using that characterization of SPEs, we prove that SPEs always exist in parity games (Theorem 25). That result had already been proved by Ummels in [14]: we use the concepts of requirements and negotiation to rephrase his proof in a more succinct way.

Main contributions

In order to get tight complexity results, we establish the links between the negotiation function and a class of zero-sum two-player games, the *abstract negotiation games* (Theorem 28). Those games are played on an infinite arena, but we show that the players can play simple strategies that have polynomial size representation, while still playing optimally (Lemma 36). We show that a non-deterministic polynomial algorithm can decide which player has a winning strategy in that game, i.e. can decide whether $\text{nego}(\lambda)(v) = 0$ or 1, for a given requirement λ on a given vertex in a given game: as a consequence, deciding whether the

requirement λ is a fixed point of the negotiation function is **NP**-easy (Lemma 38). We also show that the computation of λ^* , and consequently the SPE constrained existence problem, are fixed-parameter tractable if we fix the number of players and colors in the game (Theorem 43).

Those algorithms can be exploited to obtain upper bounds for several problems related to SPEs. Most classical of them, the problem of deciding whether there exists an SPE generating a payoff vector between two given thresholds (SPE constrained existence problem), is **NP**-complete (Theorem 48). That problem can be solved without computing the least fixed point of the negotiation function: such a problem is **BH**₂-complete, as well as its decisional version – i.e. deciding whether a given requirement λ is equal to λ^* (Theorem 49). Finally, deciding whether there exists an SPE generating a play that satisfies some LTL formula (SPE-verification problem) is **PSpace**-complete (Theorem 50).

Related works

In [9], Ummels and Grädel solve the SPE constrained existence problem in parity games, and prove that such games always contain SPEs. Their algorithm is based on the construction of alternating tree automata, on which one can solve the emptiness problem in exponential time. The SPE constrained existence problem is therefore **ExpTime**-easy, which was, to our knowledge, the best upper bound existing in the literature. The authors also prove the **NP**-hardness of that problem. In what follows, we prove that it is actually **NP**-complete.

In [8], Flesch and Predtetchinski present a general non-effective procedure to characterize the plays supported by an SPE in games with finitely many possible payoff vectors, as parity games. That characterization uses the abstract negotiation game, but does not use the notions of requirements and negotiation, and as a consequence does not yield an effective algorithm – their procedure requires to solve infinitely many games that have an uncountable state space.

In [2], we solve the SPE constrained existence problem on mean-payoff games. To that end, we define the notions of requirements and negotiation, and highlight the links between negotiation and the abstract negotiation game. Part of our results can be applied to every prefix-independent game with steady negotiation, which includes parity games. But, in order to get tight complexity results, we need here to introduce new notions, such as reduced strategies and deviation graphs.

In [4], Brihaye et al. prove that the SPE constrained existence problem on quantitative reachability games is **PSpace**-complete. Their algorithm updates continuously a function that heralds the notion of requirement, until it reaches a fixed point, that we can interpret as the least fixed point of the negotiation function.

In [5], Brihaye et al. give a characterization of NEs in cost-prefix linear games, based on the worst-case value. Parity games are cost-prefix linear, and the worst-case value is captured by our notion of requirement. The authors do not study the notion of SPE in their paper.

In [11], Meunier proposes a method to decide the existence of SPEs generating a given payoff, proving that it is equivalent to decide which player has a winning strategy in a Prover-Challenger game. That method could be used with parity games, but it would not lead to a better complexity than [14], and so it would not yield our **NP**-completeness result.

The applications of non-zero sum infinite duration games targeting reactive synthesis problems have gathered significant attention during the recent years, hence a rich literature on that topic. The interested reader may refer to the surveys [1, 6] and their references.

Structure of the paper

In Section 2, we introduce the necessary background. In Section 3, we present the concepts of requirements, negotiation and abstract negotiation game, and show how they can be applied to characterize SPEs in parity games. In Section 4, we turn that characterization into algorithms that solve the aforementioned problems, and deduce upper bounds for their complexities. In Section 5, we match them with lower bounds, and conclude on the precise complexities of those problems. The detailed proofs of our results can be found in appendices of [3], the full version of this paper.

2 Background

In the sequel, we use the word *game* for Boolean turn-based games played on finite graphs.

► **Definition 1 (Game).** A *game* is a tuple $G = (\Pi, V, (V_i)_{i \in \Pi}, E, \mu)$, where:

- Π is a finite set of *players*;
- (V, E) is a finite directed graph, whose vertices and edges are also called *states* and *transitions*, and in which every state has at least one outgoing transition;
- $(V_i)_{i \in \Pi}$ is a partition of V , where each V_i is the set of the states *controlled* by player i ;
- $\mu : V^\omega \rightarrow \{0, 1\}^\Pi$ is a *payoff function*, which maps each sequence of states ρ to the tuple $\mu(\rho) = (\mu_i(\rho))_i$ of the players' payoffs: player i *wins* ρ if $\mu_i(\rho) = 1$, and *loses* otherwise.

A play in such a game can be seen as an infinite sequence of moves of a token on the graph (V, E) : when the token is on a given vertex, the player controlling that vertex chooses the edge along which it will move. And then, the player controlling the next vertex chooses where it moves next, and so on.

► **Definition 2 (Play, history).** A *play* (resp. *history*) in the game G is an infinite (resp. finite) path in the graph (V, E) . We write $\text{Plays}G$ (resp. $\text{Hist}G$) for the set of plays (resp. histories) in G . We write Hist_iG for the set of histories in G of the form hv , where $v \in V_i$. We write $\text{Occ}(\rho)$ (resp. $\text{Occ}(h)$) for the set of vertices that occur at least once in the play ρ (resp. the history h), and $\text{Inf}(\rho)$ for the set of vertices that occur infinitely often in ρ . We write $\text{first}(h)$ (resp. $\text{first}(\rho)$) the first vertex of h (resp. ρ), and $\text{last}(h)$ its last vertex.

Often, we need to specify an initial state for a game.

► **Definition 3 (Initialized game).** An *initialized game* is a pair (G, v_0) , often written $G_{\uparrow v_0}$, where G is a game and $v_0 \in V$ is the *initial vertex*. A play (resp. history) of G is a play (resp. history) of $G_{\uparrow v_0}$ iff its first state is v_0 . We write $\text{Plays}G_{\uparrow v_0}$ (resp. $\text{Hist}G_{\uparrow v_0}$, $\text{Hist}_iG_{\uparrow v_0}$) for the set of plays (resp. histories, histories ending in V_i) in $G_{\uparrow v_0}$.

When the context is clear, we call *game* both non-initialized and initialized games.

► **Definition 4 (Strategy, strategy profile).** A *strategy* for player i in $G_{\uparrow v_0}$ is a function $\sigma_i : \text{Hist}_iG_{\uparrow v_0} \rightarrow V$ such that for each history $hv \in \text{Hist}_iG_{\uparrow v_0}$, we have $v\sigma_i(hv) \in E$.

A *strategy profile* for $P \subseteq \Pi$ is a tuple $\bar{\sigma}_P = (\sigma_i)_{i \in P}$ where each σ_i is a strategy for player i . When $P = \Pi$, the strategy profile is *complete*, and we usually write it $\bar{\sigma}$. For each $i \in \Pi$, we write $-i$ for the set $\Pi \setminus \{i\}$. When $\bar{\tau}_P, \bar{\tau}'_Q$ are two strategy profiles with $P \cap Q = \emptyset$, we write $(\bar{\tau}_P, \bar{\tau}'_Q)$ the strategy profile $\bar{\sigma}_{P \cup Q}$ defined by $\sigma_i = \tau_i$ if $i \in P$, and $\sigma_i = \tau'_i$ if $i \in Q$. We write $\Sigma_iG_{\uparrow v_0}$ (resp. $\Sigma_PG_{\uparrow v_0}$) the set of all strategies (resp. strategy profiles) for player i (resp. the set P) in $G_{\uparrow v_0}$.

A history or a play is *compatible with* (or *supported by*) a strategy σ_i if for each of its prefixes hv with $h \in \text{Hist}_i G$, we have $v = \sigma_i(h)$. It is compatible with a strategy profile $\bar{\sigma}_P$ if it is compatible with σ_i for each $i \in P$. When a strategy profile $\bar{\sigma}$ is complete, there is one unique play in $G_{\uparrow v_0}$ that is compatible with it, written $\langle \bar{\sigma} \rangle_{v_0}$ and called the *outcome* of $\bar{\sigma}$.

A strategy σ_i is *memoryless* when for each state v and every two histories h and h' , we have $\sigma_i(hv) = \sigma_i(h'v)$. In that case, we liberally consider that σ_i is defined from every state, and write $\sigma_i(v)$ for every $\sigma_i(hv)$.

Before defining the notion of SPEs, we need to define a weaker, but more classical, solution concept: Nash equilibria. A Nash equilibrium is a strategy profile such that no player can improve their payoff by deviating unilaterally from their strategy.

► **Definition 5** (Nash equilibrium). A complete strategy profile $\bar{\sigma}$ in $G_{\uparrow v_0}$ is a *Nash equilibrium* – or *NE* for short – iff for each player i and for every strategy σ'_i , we have $\mu_i(\langle \bar{\sigma}_{-i}, \sigma'_i \rangle_{v_0}) \leq \mu_i(\langle \bar{\sigma} \rangle_{v_0})$.

An SPE is an NE in all the subgames, in the following formal sense.

► **Definition 6** (Subgame, substrategy). Let hv be a history in $G_{\uparrow v_0}$. The *subgame* of G after hv is the initialized game $G_{\uparrow hv} = (\Pi, V, (V_i)_i, E, \mu_{\uparrow hv})_{\uparrow v}$, where $\mu_{\uparrow hv}$ maps each play to its payoff in G , assuming that the history hv has already been played: formally, for every $\rho \in \text{Plays}_{G_{\uparrow hv}}$, we have $\mu_{\uparrow hv}(\rho) = \mu(h\rho)$. If σ_i is a strategy in $G_{\uparrow v_0}$, its *substrategy* after hv is the strategy $\sigma_{i\uparrow hv}$ in $G_{\uparrow hv}$, defined by $\sigma_{i\uparrow hv}(h')$ for every $h' \in \text{Hist}_i G_{\uparrow hv}$.

► **Definition 7** (Subgame-perfect equilibrium). A complete strategy profile $\bar{\sigma}$ in $G_{\uparrow v_0}$ is a *subgame-perfect equilibrium* – or *SPE* for short – iff for every history hv in $G_{\uparrow v_0}$, the substrategy profile $\bar{\sigma}_{\uparrow hv}$ is a Nash equilibrium.

Throughout this paper, we mostly study *parity* games.

► **Definition 8** (Parity game). The game G is a *parity game* if there exists a tuple of *color functions* $(\kappa_i : V \rightarrow \mathbb{N})_{i \in \Pi}$, such that each play ρ is won by a given player i – i.e. $\mu_i(\rho) = 1$ – iff the least color seen infinitely often by player i , i.e. the integer $\min \kappa_i(\text{Inf}(\rho))$, is even.

A *Büchi* game is a parity game where all colors are either 0 or 1 – or equivalently, a game in which the objective of each player is to visit infinitely often a given set of vertices. A *coBüchi* game is a parity game where all colors are either 1 or 2 – or equivalently, a game in which the objective of each player is to eventually avoid a given set of vertices.

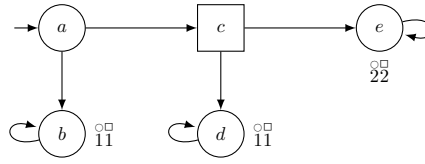
► **Example 9**. Consider the (coBüchi) game represented by Figure 1: both players win the play ace^ω , and lose any other. A first NE in that game is the strategy profile in which both players always go to the right: its outcome is ace^ω , which is won by both players, hence none can strictly improve their payoff by deviating. A second NE is the strategy profile in which both players always go down: its outcome is ab^ω , which is lost by both players. However, player \square cannot improve his strategy, because he never plays; and player \circ cannot neither, because if she goes right, then \square plans to go down, and she still loses. Only the first one is an SPE: for player \square , planning to go down from the state c is a non-credible threat.

An important property of parity games is that they are *prefix-independent*.

► **Definition 10** (Prefix-independent game). The game G is *prefix-independent* iff for every history h , we have $\mu_{\uparrow h} = \mu$ – or, equivalently, $G_{\uparrow h} = G_{\uparrow \text{last}(h)}$.

In such games, we search algorithms that solve the following problems. Let us specify that in all the sequel, tuples, as well as mappings, are ordered by the componentwise order.

10:6 On the Complexity of SPEs in Parity Games



■ **Figure 1** A coBüchi game with two NEs and one SPE.

► **Problem 11** (SPE constrained existence problem). Given a parity game $G_{\uparrow v_0}$ and two thresholds $\bar{x}, \bar{y} \in \{0, 1\}^{\Pi}$, is there an SPE $\bar{\sigma}$ in $G_{\uparrow v_0}$ such that $\bar{x} \leq \mu(\langle \bar{\sigma} \rangle_{v_0}) \leq \bar{y}$?

The next problem requires a definition of the linear temporal logic, LTL.

► **Definition 12** (LTL formulas). The *linear temporal logic* – or *LTL* for short – over the set of atomic propositions \mathbb{A} is defined as follows: syntactically, each $a \in \mathbb{A}$ is an LTL formula, and if φ and ψ are LTL formulas, then $\neg\varphi$, $\varphi \vee \psi$, $\mathbf{X}\varphi$, and $\varphi\mathbf{U}\psi$ are LTL formulas.

Semantically, if $\nu = \nu_0\nu_1\dots$ is an infinite sequence of valuations of \mathbb{A} , then:

- $\nu \models a$ iff $\nu_0(a) = 1$;
- $\nu \models \neg\varphi$ iff $\nu \not\models \varphi$;
- $\nu \models \varphi \vee \psi$ iff $\nu \models \varphi$ or $\nu \models \psi$;
- $\nu \models \mathbf{X}\varphi$ iff $\nu_1\nu_2\dots \models \varphi$;
- $\nu \models \varphi\mathbf{U}\psi$ iff there exists $k \in \mathbb{N}$ such that $\nu_k\nu_{k+1}\dots \models \psi$, and for each $\ell < k$, we have $\nu_\ell\nu_{\ell+1}\dots \models \varphi$.

We will also make use of the classical notations $\wedge, \top, \perp, \Rightarrow, \mathbf{F}$ or \mathbf{G} defined as abbreviations using the symbols chosen here as primitives. In particular, we write \top for $a \vee \neg a$, $\mathbf{F}\varphi$ (“finally φ ”) for $\top\mathbf{U}\varphi$, and $\mathbf{G}\varphi$ (“globally φ ”) for $\neg\mathbf{F}\neg\varphi$. When we use LTL to describe plays in a game, w.l.o.g. and for simplicity, the atom set is $\mathbb{A} = V$, and each play ρ is assimilated to the sequence of valuations ν defined by $\nu_k(v) = 1$ iff $\rho_k = v$. For example, when u and v are two vertices, the play $(uv)^\omega$ is the only play satisfying the formula $u \wedge \mathbf{G}((u \Rightarrow \mathbf{X}v) \wedge (v \Rightarrow \mathbf{X}u))$.

► **Problem 13** (SPE-verification problem). Given a parity game $G_{\uparrow v_0}$ and an LTL formula φ , is there an SPE $\bar{\sigma}$ in $G_{\uparrow v_0}$ such that $\langle \bar{\sigma} \rangle_{v_0} \models \varphi$?

► **Remark.** The natural problems of deciding whether *all* SPEs generate a payoff vector between two thresholds, or a play that satisfies some LTL formula, are the duals of the aforementioned problems, and their complexities are obtained as direct corollaries. For example, in a given parity game, all the outcomes of SPEs satisfy the formula φ if and only if there does not exist an SPE whose outcome satisfies $\neg\varphi$. Since we will show that the SPE-verification problem is **PSpace**-complete, its dual will also be **PSpace**-complete. Similarly, the SPE constrained universality problem is **coNP**-complete as we will show that its dual, the SPE constrained existence problem, is **NP**-complete.

While we do not recall the definition of classical complexity classes here (such as **NP**, **coNP**, or **PSpace**, see [12]), we recall the definition of the class **BH₂**: the second level of the Boolean hierarchy. For more details about the Boolean hierarchy itself, see [15].

► **Definition 14** (Class **BH₂**). The complexity class **BH₂** is the class of problems of the form $P \cap Q$, where P is **NP**-easy and Q is **coNP**-easy, and both have the same set of instances. In other words, a **BH₂**-easy problem is a problem that can be decided with one call to an **NP** algorithm, and one to a **coNP** algorithm.

► **Remark.** The class \mathbf{BH}_2 must not be mistaken with the class $\mathbf{NP} \cap \mathbf{coNP}$, that gathers the problems that can be solved by an \mathbf{NP} algorithm *as well as* by a \mathbf{coNP} one: the latter is included in \mathbf{NP} and in \mathbf{coNP} , while the former contains them.

To attach intuition to this definition, let us present a useful \mathbf{BH}_2 -complete problem.

► **Problem 15** ($\mathbf{SAT} \times \mathbf{COSAT}$). Given a pair (φ_1, φ_2) of propositional logic formulas, is it true that φ_1 is satisfiable and that φ_2 is not?

► **Lemma 16.** *The problem $\mathbf{SAT} \times \mathbf{COSAT}$ is \mathbf{BH}_2 -complete.*

3 Negotiation in parity games

3.1 Requirements, negotiation, and link with SPEs

In the algorithms we provide, we make use of the characterization of SPEs in prefix-independent games that has been presented in [2]. We recall here the notions needed, slightly adapted to Boolean games.

► **Definition 17** (Requirement). A *requirement* on a game G is a mapping $\lambda : V \rightarrow \{0, 1, +\infty\}$. The set of the requirements on G is denoted by $\mathbf{Req}G$, and is ordered by the componentwise order \leq : we write $\lambda \leq \lambda'$ when we have $\lambda(v) \leq \lambda'(v)$ for all v .

► **Definition 18** (λ -consistency). Let λ be a requirement on the game G . A play ρ in G is λ -consistent iff for each player i and every index k such that $\rho_k \in V_i$, we have $\mu_i(\rho_k \rho_{k+1} \dots) \geq \lambda(\rho_k)$. The set of λ -consistent plays in $G_{\uparrow v_0}$ is denoted by $\lambda\mathbf{Cons}(v_0)$.

In this paper, we will consider specifically requirements that are *satisfiable*.

► **Definition 19** (Satisfiability). The requirement λ , on the game G , is *satisfiable* iff for each state $v \in V$, there exists at least one λ -consistent play from v .

► **Lemma 20.** *Given a parity game G and a requirement λ , deciding whether λ is satisfiable is \mathbf{NP} -easy.¹*

Proof. An \mathbf{NP} algorithm for that problem guesses, first, a family $(h_v, W_v)_{v \in V}$, where for each v , h_v is a history without cycle starting from v and W_v is a subset of V with $\mathbf{last}(h_v) \in W_v$. That family is an object of polynomial size, and certifies that λ is satisfiable if for each v : (1) $\lambda(v) \neq +\infty$; (2) the subgraph $(W_v, E \cap W_v^2)$ is strongly connected; (3) for each vertex $u \in \mathbf{Occ}(h_v) \cup W_v$ such that $\lambda(u) = 1$, if i is the player who controls u , then the color $\min \kappa_i(W_v)$ is even.

Indeed, if those three points are satisfied, then for each v , the play $h_v c_v^\omega$, where c_v is a cycle (not necessarily simple, but which can be chosen of size at most $(\mathbf{card}W_v)^2$) that visits all the vertices of W_v at least once and none other, is λ -consistent. Conversely, if from each v , there exists a λ -consistent play ρ , then the pair (h_v, W_v) , where $W_v = \mathbf{Inf}(\rho)$ and h_v is a prefix of ρ ending in W_v in which the cycles have been removed, satisfies those three properties – those can be checked in polynomial time. ◀

Each requirement induces a notion of rationality for a coalition of players.

¹ It is actually \mathbf{NP} -complete, as we can prove by slightly adapting the proof of Theorem 47.

► **Definition 21** (λ -rationality). Let λ be a requirement on the game G . A strategy profile $\bar{\sigma}_{-i}$ is λ -rational assuming the strategy σ_i iff for every history hv compatible with $\bar{\sigma}_{-i}$, the play $\langle \bar{\sigma}_{\uparrow hv} \rangle_v$ is λ -consistent. It is λ -rational if it is λ -rational assuming some strategy. The set of λ -rational strategy profiles in $G_{\uparrow v_0}$ is denoted by $\lambda\text{Rat}(v_0)$.

The notion of λ -rationality qualifies the *environment* against player i , i.e. the coalition of all the players except i : they play λ -rationally if their strategy profile can be completed by a strategy of player i , such that in every subgame, each player gets their requirement satisfied.

But then, λ -rationality restrains the behaviours of the players against player i : that one may be able to win against a λ -rational environment while it is not the case against a fully hostile one. This is what the *negotiation function* captures.

► **Definition 22** (Negotiation). The *negotiation function* is a function that transforms every requirement λ into a requirement $\text{nego}(\lambda)$, defined by, for each $i \in \Pi$ and $v \in V_i$, and with the convention $\inf \emptyset = +\infty$:

$$\text{nego}(\lambda)(v) = \inf_{\bar{\sigma}_{-i} \in \lambda\text{Rat}(v)} \sup_{\sigma_i \in \Sigma_i(G_{\uparrow v})} \mu_i(\langle \bar{\sigma}_{-i}, \sigma_i \rangle_v).$$

► **Remark.** If player i follows the strategy σ_i assuming which $\bar{\sigma}_{-i}$ is λ -rational, then they get at least the payoff $\lambda(v)$, hence $\text{nego}(\lambda)(v) \geq \lambda(v)$ and the negotiation function is non-decreasing. Moreover, if $\lambda \leq \lambda'$, then all the λ' -rational strategy profiles are also λ -rational, hence $\text{nego}(\lambda) \leq \text{nego}(\lambda')$ and the negotiation function is monotonic.

The fixed points of the negotiation function characterize the SPEs of a game: indeed, when some play ρ is λ -consistent for some fixed point λ , it means that it is won by every player who could ensure their victory from a state visited by ρ , while playing against a rational environment. Better: all the SPEs are characterized by the least fixed point of the negotiation function, which exists by Tarski's fixed point theorem, and which we will write λ^* in the rest of this paper. An equivalent result exists for NEs, that are characterized by the requirement $\text{nego}(\lambda_0)$, where $\lambda_0 : v \mapsto 0$ is the *vacuous requirement*.

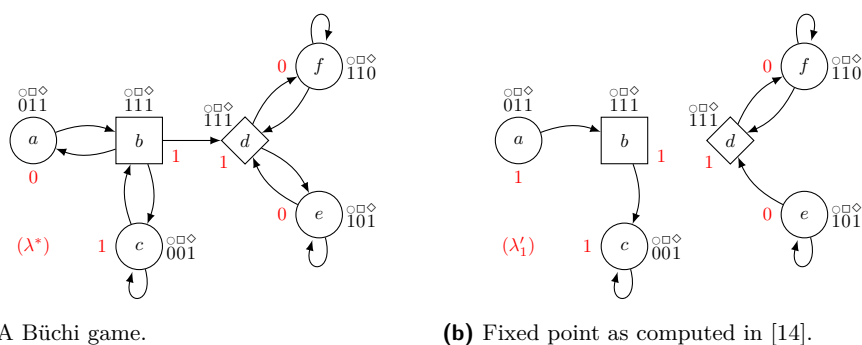
► **Theorem 23.** In a prefix-independent Boolean game $G_{\uparrow v_0}$:

- the set of NE outcomes is exactly the set of $\text{nego}(\lambda_0)$ -consistent plays;
- the set of SPE outcomes is exactly the set of λ^* -consistent plays.

Proof. By [2], this result is true for any prefix-independent game *with steady negotiation*, i.e. such that for every requirement λ , for every player i and for every vertex v , if there exists a λ -rational strategy profile $\bar{\sigma}_{-i}$ from v , there exists one that minimizes the quantity $\sup_{\sigma_i} \mu_i(\langle \bar{\sigma}_{-i}, \sigma_i \rangle_v)$. In the case of Boolean games, the function μ_i can only take the values 0 and 1, hence this supremum is always realized. ◀

► **Example 24.** Let us consider again the game of Figure 1. Every play in that game – like in every game – is λ_0 -rational. The requirement $\lambda_1 = \text{nego}(\lambda_0)$ is equal to 1 on the states c and e (the states from which the player controlling those states can enforce the victory), and to 0 in each other one. Then, the λ_1 -consistent plays are exactly the plays supported by a Nash equilibrium: the play ace^ω , and the play ab^ω .

Now, from the state a , the only strategy profile that can make player \circ lose if she chooses to go to c is $\sigma_\square : ac \mapsto d$, which was λ_0 -rational but is not λ_1 -rational: the play cd^ω is not λ_1 -consistent. Therefore, against a λ_1 -rational environment, player \circ can enforce the victory by going to the state c , hence $\lambda_2(a) = 1$, where $\lambda_2 = \text{nego}(\lambda_1)$. Then, the requirement λ_2 is a fixed point of the negotiation function, and consequently the least one, hence the only play supported by an SPE from the state a is the only play that is λ_2 -consistent, namely ace^ω .



■ **Figure 2** An illustration of Ummels' algorithm.

3.2 The existence of SPEs in parity games

In parity games, the existence of SPEs is guaranteed.

► **Theorem 25** ([14]). *There exists an SPE in every parity game.*

Proof sketch. This theorem is a result due to Ummels. In [3], we rephrase his proof in terms of requirements and negotiation. Let us give here the main intuitions. We define a decreasing sequence $(E_n)_n$ of subsets of E , and an associated sequence $(\lambda'_n)_n$ of requirements, keeping the hypothesis that E_n always contains at least one outgoing edge from each vertex. First, $E_0 = E$ and λ'_0 is the vacuous requirement. Then, for every n , for each player i and each $v \in V_i$, we define $\lambda'_{n+1}(v)$ as equal to 1 if and only if in the game obtained from G by removing the edges that are not in E_n , player i can enforce the victory from v , against a fully hostile environment. Then, from each such state, we choose a memoryless winning strategy (which always exists, see [10]), that is, we choose one edge to always follow to ensure the victory, and we remove the other outgoing edges from E_n to obtain E_{n+1} . We prove that for each n , we have $\lambda'_{n+1} \geq \text{nego}(\lambda'_n)$, hence the sequence $(\lambda'_n)_n$ converges to a satisfiable fixed point of the negotiation function – which is not necessarily the least one. ◀

As a consequence, in every parity game G , we have $\lambda^*(v) \in \{0, 1\}$: this is why in what follows, we only consider requirements with values in $\{0, 1\}$.

► **Example 26.** Let us consider the (Büchi) game of Figure 2a: recall that the objective of each player is to see infinitely often the color 0. If we follow the algorithm from [14], as presented above, we remove the edge df – because always going to e is a winning strategy for player \diamond from d – and the edges ba and bd – because always going to c is a winning strategy for player \square from b . Then, the algorithm reaches a fixed point, see Figure 2b. Our proof states that every play that uses only the remaining edges is a play supported by an SPE. Indeed, those plays are λ'_1 -consistent, where λ'_1 is the requirement given in red on Figure 2b: it is a fixed point of the negotiation function. However, it is not the least one: for example, the play $ab(dedf)^\omega$ is not λ'_1 -consistent, but it is also a play that is supported by an SPE. The least fixed point is given in red on Figure 2a.

The interested reader will find in [3] an additional example of parity game, on which we computed the iterations of the negotiation function.

3.3 Abstract negotiation game

Now, let us study how we can compute the negotiation function. The *abstract negotiation game* is a tool which already appeared in [8], and which has been linked to the negotiation function in [2]. It is a game on an infinite graph that opposes two players, *Prover* and

10:10 On the Complexity of SPEs in Parity Games

Challenger: Prover constructs a λ -rational strategy profile by proposing plays, and Challenger constructs player i 's response by accepting those plays or deviating from them. We slightly simplify the definition here, by considering only satisfiable requirements, which guarantees that Prover has always a play to propose².

► **Definition 27** (Abstract negotiation game). Let G be a parity game, let λ be a satisfiable requirement, let $i \in \Pi$ and let $v_0 \in V_i$. The associated *abstract negotiation game* is the two-player zero-sum game $\text{Abs}_{\lambda i}(G)_{\uparrow[v_0]} = (\{\mathbb{P}, \mathbb{C}\}, S, (S_{\mathbb{P}}, S_{\mathbb{C}}), \Delta, \nu)_{\uparrow[v_0]}$ where:

- the players \mathbb{P} and \mathbb{C} are called respectively *Prover* and *Challenger*;
- Challenger's states are of the form $[\rho]$, where ρ is a λ -consistent play of G ;
- Prover's states are of the form $[hv]$, where $h \in \text{Hist}_i(G) \cup \{\varepsilon\}$ and $\text{last}(h)v \in E$, plus one additional sink state \top ;
- the set Δ contains the transitions of the forms:
 - $[v][\rho]$, where $\text{first}(\rho) = v$: Prover proposes the play ρ ;
 - $[\rho][\rho_0 \dots \rho_k v]$, where $k \in \mathbb{N}$, $v \neq \rho_{k+1}$ and $\rho_k v \in E$: Challenger refuses and deviates;
 - $[hv][v]$ with $h \neq \varepsilon$: then, Prover has to propose a new play from the vertex v ;
 - $[\rho]\top$: Challenger accepts the proposed play;
 - $\top\top$: the game is over;
- When π is a play in the abstract negotiation game, we will use the notation $\hat{\pi}$ to denote the play in the original game constructed by Prover's proposals and Challenger's deviations. Thus, the play π is won by Challenger iff one of the following conditions is satisfied:
 - the play π has the form $[v_0][\rho^0][h^0 v_1][v_1][\rho^1] \dots [h^{n-1} v_n][v_n][\rho^n]\top^\omega$, i.e. Challenger accepts a play proposed by Prover, and the play $\hat{\pi} = h^0 \dots h^{n-1} \rho^n$ is won by player i ;
 - or the play π has the form $[v_0][\rho^0][h^0 v_1][v_1][\rho^1][h^1 v_2] \dots$, i.e. Challenger always deviates from the play proposed by Prover, and the play $\hat{\pi} = h^0 h^1 \dots$ is won by player i .

► **Theorem 28** ([2], Appendix E). *Let G be a prefix-independent Boolean game, let λ be a satisfiable requirement, let $i \in \Pi$ and let $v_0 \in V_i$. Then, we have $\text{nego}(\lambda)(v_0) = 0$ if and only if Prover has a winning strategy in the associated abstract negotiation game.*³

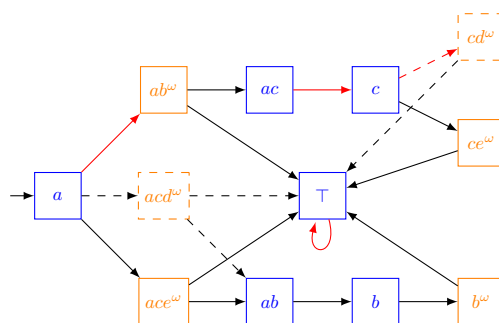
► **Example 29**. Let G be the game from Figure 1. In this particular case, since there are finitely many possible plays, the abstract negotiation games $\text{Abs}_{\lambda_0 \circ} G$ and $\text{Abs}_{\lambda_1 \circ} G$ have a finite state space. They are represented in Figure 3: the blue states are Prover's states, and the orange ones are Challenger's. The dashed states belong to $\text{Abs}_{\lambda_0 \circ} G$ but not to $\text{Abs}_{\lambda_1 \circ} G$. Observe that Prover has a winning strategy in $\text{Abs}_{\lambda_0 \circ} G$ (in red), but not in $\text{Abs}_{\lambda_1 \circ} G$.

4 Algorithms

Let us now study how we can use the abstract negotiation game to solve the problems presented in the introduction. We first define an equivalence relation between histories and between plays; then, we show that in the abstract negotiation game, Prover can propose only plays that are simple representatives of their equivalence class, and propose always the same play from each vertex.

² In [2], a second sink state \perp is added to enable Prover to give up when she has no play to propose. Another purely technical difference is the existence here of a mandatory transition from each state $[hv]$ to the state $[v]$, instead of letting Prover propose a play directly from the state $[hv]$: thus, there are few states from which Prover has a choice to make, which will be useful in what follows.

³ The non-existence of the sink state \perp , in which Prover is supposed to get the payoff $-\infty$, does not change this result: since λ is assumed to be satisfiable, Prover has always a strategy to get at least the payoff 0, hence no optimal strategy of Prover plans to follow a transition to \perp .



■ **Figure 3** An abstract negotiation game.

4.1 Reduced plays and reduced strategy

The equivalence relation that we use is based on the order in which vertices appear.

► **Definition 30** (Occurrence-equivalence (histories)). Two histories h and h' are *occurrence-equivalent*, written $h \approx h'$, iff $\text{first}(h) = \text{first}(h')$, $\text{last}(h) = \text{last}(h')$ and $\text{Occ}(h) = \text{Occ}(h')$.

► **Definition 31** (Occurrence-equivalence (plays)). Two plays ρ and ρ' are *occurrence-equivalent*, written $\rho \approx \rho'$, iff the three following conditions are satisfied:

- $\text{Inf}(\rho) = \text{Inf}(\rho')$;
- for each history prefix of ρ , there exists a occurrence-equivalent history prefix of ρ' ;
- for each history prefix of ρ' , there exists a occurrence-equivalent history prefix of ρ .

► **Example 32.** Let us consider the game of Figure 2a. In that game, the play $ab(\text{dedf})^\omega$ is occurrence-equivalent to the play $abde(\text{dedf})^\omega$, but not to the play $ab(\text{dfde})^\omega$. Indeed, the latter has the history $abdf$ as a prefix, which is not occurrence-equivalent to any prefix of $ab(\text{dedf})^\omega$, in which the state f occurs only when the state e has already occurred.

► **Remark.** The operators Occ , Inf and μ are stable by occurrence-equivalence.

The interest of that equivalence relation lies in the finite number of its equivalence classes, and by the existence of simple representatives of each of them.

► **Lemma 33.** *Let ρ be a play of G . There exists a lasso $hc^\omega \approx \rho$ with $|h| \leq n^3 + n^2$ and $|c| \leq n^2$, where $n = \text{card}V$.*

We call such lassos *reduced plays*. For each ρ , we write $\tilde{\rho}$ for an arbitrary occurrence-equivalent reduced play. Then, operations such as computing $\mu(\tilde{\rho})$, $\text{Occ}(\tilde{\rho})$, $\text{Inf}(\tilde{\rho})$, or checking whether $\tilde{\rho}$ is λ -consistent, can be done in time $O(n^3)$.

► **Definition 34** (Reduced strategy). A strategy $\tau_{\mathbb{P}}$ for Prover in $\text{Abs}_{\lambda_i}(G)$ is *reduced* iff it is memoryless, and for each state v , the play ρ with $[\rho] = \tau_{\mathbb{P}}([v])$ is a reduced play.

► **Example 35.** In Figure 3, Prover's winning strategy, defined by the red arrows, is reduced.

If $\rho \approx \rho'$, and if Challenger can deviate from ρ after the history hv , then he can also deviate in ρ' after some history $h'v$ that traverses the same states. Thus, Prover can play optimally while proposing only reduced plays, and by proposing always the same play from each vertex; that is, by following a reduced strategy.

► **Lemma 36.** *Prover has a winning strategy in the abstract negotiation game if and only if she has a reduced one.*

4.2 Checking that a reduced strategy is winning: the deviation graph

We have established that Prover is winning the abstract negotiation game if and only if she has a reduced winning strategy. Such a strategy has polynomial size and can thus be guessed in nondeterministic polynomial time. It remains us to show that we can verify in deterministic polynomial time that a guessed strategy is winning. For that purpose, we construct its *deviation graph*.

► **Definition 37** (Deviation graph). Let $\tau_{\mathbb{P}}$ be a reduced strategy of Prover, and let $i \in \Pi$. The *deviation graph* associated to $\tau_{\mathbb{P}}$ and v is the colored graph $\text{Dev}_i(\tau_{\mathbb{P}})$:

- the vertices are the plays $\tau_{\mathbb{P}}([w])$, for every vertex w of the original game;
- there is an edge from $[\tilde{\rho}]$ to $\tau_{\mathbb{P}}([w])$ with color c iff there exists $k \in \mathbb{N}$ such that $\tilde{\rho}_k \in V_i$, $\tilde{\rho}_k w \in E$, $w \neq \tilde{\rho}_{k+1}$ and $\min \kappa_i(\text{Occ}(\tilde{\rho}_0 \dots \tilde{\rho}_k)) = c$.

Constructing the deviation graph associated to a memoryless strategy $\tau_{\mathbb{P}}$ enables to decide whether $\tau_{\mathbb{P}}$ is a winning strategy or not.

► **Lemma 38.** *The reduced strategy $\tau_{\mathbb{P}}$ is winning in the abstract negotiation game if and only if in the corresponding deviation graph, there neither exists, from the vertex $\tau_{\mathbb{P}}([v_0])$:*

- a finite path to a vertex $[\tilde{\rho}]$ such that the play $\tilde{\rho}$ is winning for player i ;
- nor an infinite path along which the minimal color seen infinitely often is even.

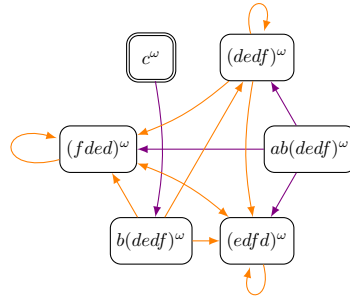
*As a consequence, given a parity game G and a requirement λ , deciding whether λ is a fixed point of the negotiation function is **NP**-easy.*

Proof. Given G and λ , let n be the number of states in G , and m be the number of colors. The deviation graph can be seen as the abstract negotiation game itself, where one removed the transitions that were not compatible with $\tau_{\mathbb{P}}$; removed the states that were not accessible from $[v_0]$; and merged the paths $[\tilde{\rho}][hv][v][\tilde{\rho}']$ into one edge $[\tilde{\rho}][\tilde{\rho}']$ with color $\min \kappa_i(h)$.

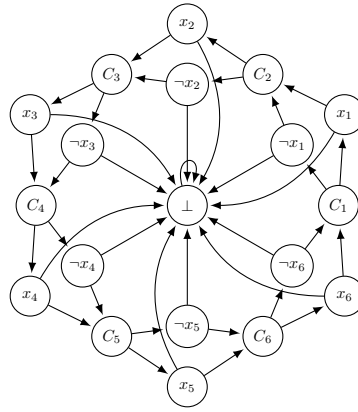
Therefore, a path from the vertex $\tau_{\mathbb{P}}([v_0])$ can be seen as a history (if it is finite) or a play (if it is infinite), compatible with the strategy $\tau_{\mathbb{P}}$, in the abstract negotiation game. In particular, the finite paths to a vertex $[\tilde{\rho}]$ with $\mu_i(\tilde{\rho}) = 1$ correspond to the histories that lead Prover to propose a play that is winning for player i , that Challenger can accept to win. Similarly, the infinite paths along which the minimal color seen infinitely often is even correspond to the plays π where Challenger deviates infinitely often, and constructs the play $\tilde{\pi}$ that is winning for player i . Such paths will be called *winning paths*.

Now, the deviation graph has n vertices, and at most mn^2 edges. Constructing it requires time $O(n^4)$. Deciding the existence of a finite winning path is a simple accessibility problem, and can be done in time $O(mn^2)$. Deciding the existence of an infinite winning path is similar to deciding the emptiness of a parity automaton, and requires a time $O(mn^3)$. As a consequence, deciding whether a reduced strategy is winning can be done in polynomial time – and it is an object of polynomial size.

Thus, an **NP** algorithm that decides whether the requirement λ is a fixed point of the negotiation function is the following: we guess, at the same time, a certificate that proves that λ is satisfiable (Lemma 20), and a reduced strategy $\tau_{\mathbb{P}}^v$ for Prover in $\text{Abs}_{\lambda_i}(G)_{[v]}$, for each $i \in \Pi$ and $v \in V_i$ such that $\lambda(v) = 0$: by Lemma 36, there exists such a winning strategy if and only if $\text{nego}(\lambda)(v) = 0$. Those objects form a certificate of polynomial size, that can be checked in polynomial time. ◀



■ **Figure 4** A deviation graph.



■ **Figure 5** The game G_φ .

► **Example 39.** Let us consider the game of Figure 2a, and the requirement λ^* , presented on the same figure. Let $\tau_{\mathbb{P}}$ be the memoryless strategy in the game $\text{Abs}_{\lambda^* \circ}(G)_{\upharpoonright [a]}$ defined by:

$$\begin{aligned} \tau_{\mathbb{P}}([a]) &= ab(dedf)^\omega & \tau_{\mathbb{P}}([c]) &= c^\omega & \tau_{\mathbb{P}}([e]) &= (edfd)^\omega \\ \tau_{\mathbb{P}}([b]) &= b(dedf)^\omega & \tau_{\mathbb{P}}([d]) &= (dedf)^\omega & \tau_{\mathbb{P}}([f]) &= (fded)^\omega. \end{aligned}$$

The corresponding deviation graph is given by Figure 4. The purple edges have color 0, and the orange ones have color 1. Observe that there is no winning path from the vertex $\tau_{\mathbb{P}}([a]) = ab(dedf)^\omega$: each purple edge can be used at most once, and even though the play c^ω is winning for player \circ , the corresponding vertex is not accessible. Therefore, the strategy $\tau_{\mathbb{P}}$ is winning in $\text{Abs}_{\lambda^* \circ}(G)_{\upharpoonright [a]}$, which proves the equality $\text{nego}(\lambda^*)(a) = \lambda^*(a) = 0$.

4.3 Upper bounds

Let us now give the main problems for which the concepts given above yield a solution.

A first application is an algorithm for the SPE constrained existence problem.

► **Lemma 40.** *The SPE constrained existence problem for parity games is NP-easy.*

Proof. Given a parity game $G_{\upharpoonright v_0}$ and two thresholds \bar{x} and \bar{y} , we can guess a reduced play $\tilde{\eta}$ from v_0 , a requirement λ , and the certificates required to decide whether λ is a fixed point of nego , according to Lemma 38. All those objects have polynomial size. Then, to check that $\tilde{\eta}$ is an SPE outcome, using Theorem 23, we check that λ is a fixed point of nego , that $\tilde{\eta}$ is λ -consistent, and that $\bar{x} \leq \mu(\tilde{\eta}) \leq \bar{y}$ in polynomial time. ◀

10:14 On the Complexity of SPEs in Parity Games

This algorithm does not need an effective characterization of all the SPEs in a game, which is given by the least fixed point of the negotiation function λ^* . Computing such a characterization can be done with a call to a **NP** oracle and to a **coNP** oracle, i.e. it belongs to the class **BH**₂.

► **Lemma 41.** *Given a parity game G and a requirement λ , deciding whether $\lambda = \lambda^*$ is **BH**₂-easy.*

Proof. First, deciding whether λ is a fixed point of the negotiation function is an **NP**-easy problem by Lemma 38. Deciding whether it is the least one is **coNP**-easy, because a negative instance can be recognized as follows. We guess a requirement $\lambda' < \lambda$, and the certificates of the algorithm given by Lemma 38; then, we check that λ' is a fixed point of nego. ◀

Finally, SPE-verification requires polynomial space.

► **Lemma 42.** *The SPE-verification problem is **PSpace**-easy.*

Proof. Given a parity game $G_{\uparrow v_0}$ and an LTL formula φ , by Lemma 41, the requirement λ^* can be computed by a deterministic algorithm using polynomial space – indeed, we have the inclusions **NP** \subseteq **PSpace** and **coNP** \subseteq **PSpace**, hence the guess of λ^* , followed by one call to an **NP** algorithm and one call to a **coNP** algorithm, can be transformed into a **PSpace** algorithm. Then, we can construct in polynomial time the LTL formula ψ_{λ^*} , that is satisfied exactly by the λ^* -consistent plays:

$$\psi_{\lambda^*} = \bigwedge_i \bigwedge_{v \in V_i, \lambda^*(v)=1} \left(\mathbf{F}v \Rightarrow \bigvee_{2k \leq m} \left(\bigvee_{\kappa_i(w)=2k} \mathbf{GF}w \wedge \bigvee_{\kappa_i(w) < 2k} \mathbf{FG}\neg w \right) \right),$$

where m is the largest color appearing in G .

Then, deciding whether there exists an SPE outcome in $G_{\uparrow v_0}$ that satisfies the formula φ is equivalent to decide whether there exists a play in $G_{\uparrow v_0}$ that satisfies the formula $\varphi \wedge \psi_{\lambda^*}$. As for any LTL formula, that can be done using polynomial space: see for example [13]. ◀

4.4 Fixed-parameter tractability

We end this section by mentioning an additional complexity result on the SPE constrained existence problem: it is fixed-parameter tractable.

► **Theorem 43.** *The SPE constrained existence problem on parity games is fixed-parameter tractable when the number of players and the number of colors are parameters. More precisely, there exists a deterministic algorithm that solves that problem in time $O(2^{2^{pm}} n^{12})$, where n is the number of vertices, p is the number of players and m is the number of colors.*

Proof sketch. This result is obtained by constructing and solving a generalized parity game on a finite arena, that is exponential in the number of players and polynomial in the size of the original game. This game, called the *concrete* negotiation game, is equivalent to the abstract negotiation game. Its construction is inspired by a construction that we have introduced in [2], for games with mean-payoff objectives. The main idea of this construction is to decompose the plays proposed by Prover, by passing them to Challenger edges by edges, and by encoding the λ -consistence condition into a generalized parity condition. Then, we can apply a FPT algorithm to solve generalized parity games that was first proposed in [7]. While this deterministic algorithm does not improve on the worst-case complexity of the deterministic **ExpTime** algorithm of [14], it allows for a finer parametric analysis. ◀

5 Matching lower bounds

In this section, we provide matching complexity lower bounds for the problems we addressed in the previous section. For that purpose, we first need the following construction, inspired from a game designed by Ummels in [9] to prove the **NP**-hardness of the SPE constrained existence problem. Our definition is slightly different: while Ummels defined one player per variable, we need one player per literal. However, the core intuitions are the same.

► **Definition 44** (G_φ). Let $\varphi = \bigwedge_{j \in \mathbb{Z}/m\mathbb{Z}} C_j$ be a formula of the propositional logic, constructed on the finite set of variables $\{x_1, \dots, x_n\}$. We define the parity game G_φ as follows.

- The players are the variables x_1, \dots, x_n , their negations, and *Solver*, denoted by \mathbb{S} .
- The states controlled by Solver are all the clauses C_j , and the sink state \perp .
- The states controlled by player $L = \pm x_i$ are the pairs (C_j, L) , where L is a literal of C_j .
- There are edges from each clause state C_j to all the states (C_j, L) ; from each pair state (C_j, L) to the state C_{j+1} , and to the sink state \perp ; and from the sink state \perp to itself.
- For Solver, every state has the color $\kappa_{\mathbb{S}}(v) = 2$, except the state \perp , which has color 1.
- For each literal player L , every state has the color $\kappa_L(v) = 2$, except the states of the form (C, \bar{L}) , that have the color 1.

► **Remark.** The game G_φ is a coBüchi game: Solver has to avoid the sink state \perp , and player L the states of the form (C, \bar{L}) . Therefore, all the following theorems can also be applied to the more restrictive class of coBüchi games.

► **Example 45.** The game G_φ , when φ is the tautology $\bigwedge_{j=1}^6 (x_j \vee \neg x_j)$, is given by Figure 5.

The game G_φ is strongly linked with the satisfiability of φ , in the following formal sense.

► **Lemma 46.** *The game G_φ has the following properties.*

- *The least fixed point of the negotiation function is equal to 0 on the states controlled by Solver, and to 1 on the other ones.*
- *For every SPE outcome ρ in G_φ that does not reach \perp , the formula φ is satisfied by:*

$$\nu_\rho : x \mapsto \begin{cases} 1 & \text{if } \exists C, (C, x) \in \text{Inf}(\rho) \\ 0 & \text{otherwise.} \end{cases}$$

- *Conversely, for every valuation ν satisfying φ , the play $\rho_\nu = (C_1(C_1, L_1) \dots C_m(C_m, L_m))^\omega$, where for each j , the literal L_j is satisfied by ν , is an SPE outcome.*

A first consequence is the lower bound on the complexity of deciding whether some requirement is, or not, a fixed point of the negotiation function.

► **Theorem 47.** *Given a parity game G and a requirement λ , deciding whether λ is a fixed point of the negotiation function is **NP**-complete.*

Proof. Easiness is given by Lemma 38. For hardness, we proceed by reduction from the **NP**-complete problem SAT. Given a formula $\varphi = \bigwedge_{j \in \mathbb{Z}/m\mathbb{Z}} C_j$ of the propositional logic, we can construct the game G_φ in polynomial time. Then, let us define on G_φ the requirement λ that is constantly equal to 1, except in \perp , where it is equal to 0. Since there is no winning play for Solver from \perp , the requirement λ is a fixed point of the negotiation function if and only if there exists a λ -consistent play from each vertex. If it is the case, then let ρ be a

10:16 On the Complexity of SPEs in Parity Games

λ -consistent play from C_1 . Since $\lambda \geq \lambda^*$, the play ρ is also λ^* -consistent, and is therefore an SPE outcome. It is also a play won by Solver, since $\lambda(C_1) = 1$: therefore, it does not reach the state \perp . Then, by Lemma 46, we can define the valuation ν_ρ , which satisfies φ .

Conversely, if ν is a valuation satisfying φ , then the play ρ_ν is an SPE outcome from C_1 , and it does not end in \perp , hence it is won by Solver, and is therefore λ -consistent. If we consider the suffixes of ρ_ν , we find a λ^* -consistent play from each state C_j ; and from the states of the form (C_j, L) , the play $(C_j, L)\perp^\omega$ is λ^* -consistent, hence there is a λ^* -consistent play from each vertex: deciding whether λ is a fixed point of the negotiation function is NP-hard. \blacktriangleleft

A similar proof ensures the same lower bound on the SPE constrained existence problem.

► **Theorem 48.** *The SPE constrained existence problem in parity games is NP-complete.*

Proof. Easiness is given by Lemma 40. For hardness, we proceed by reduction from the problem SAT. Given a formula $\varphi = \bigwedge_{j \in \mathbb{Z}/m\mathbb{Z}} C_j$ of the propositional logic, we can construct the game G_φ in polynomial time. By Lemma 46, there exists an SPE outcome from C_1 and won by Solver, i.e. an SPE outcome from C_1 that does not reach the sink state \perp , if and only if there exists a valuation satisfying φ : the SPE constrained existence problem is NP-hard. \blacktriangleleft

Now, we show that the algorithm we presented to compute λ^* is optimal as well.

► **Theorem 49.** *Given a parity game G and a requirement λ , deciding whether $\lambda = \lambda^*$ is \mathbf{BH}_2 -complete. Given a parity game G , computing λ^* can be done by a non-deterministic algorithm in polynomial time iff $\mathbf{NP} = \mathbf{BH}_2$.*

Finally, the LTL model-checking problem easily reduces to the SPE-verification problem, which gives us the matching lower bound.

► **Theorem 50.** *The SPE-verification problem is PSpace-complete.*

References

- 1 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 *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.
- 2 Léonard Brice, Jean-François Raskin, and Marie van den Bogaard. Subgame-perfect equilibria in mean-payoff games. *CoRR*, abs/2101.10685, 2021. [arXiv:2101.10685](https://arxiv.org/abs/2101.10685).
- 3 Léonard Brice, Marie van den Bogaard, and Jean-François Raskin. On the complexity of spes in parity games. *CoRR*, abs/2107.07458, 2021. [arXiv:2107.07458](https://arxiv.org/abs/2107.07458).
- 4 Thomas Brihaye, Véronique Bruyère, Aline Goeminne, Jean-François Raskin, and Marie van den Bogaard. The complexity of subgame perfect equilibria in quantitative reachability games. In *CONCUR*, volume 140 of *LIPICs*, pages 13:1–13:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.
- 5 Thomas Brihaye, Julie De Pril, and Sven Schewe. Multiplayer cost games with simple nash equilibria. In *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.

- 6 Véronique Bruyère. Computer aided synthesis: A game-theoretic approach. In *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.
- 7 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.
- 8 János Flesch and Arkadi Predtetchinski. A characterization of subgame-perfect equilibrium plays in borel games of perfect information. *Math. Oper. Res.*, 42(4):1162–1179, 2017. doi:10.1287/moor.2016.0843.
- 9 Erich Grädel and Michael Ummels. Solution Concepts and Algorithms for Infinite Multiplayer Games. In Krzysztof Apt and Robert van Rooij, editors, *New Perspectives on Games and Interaction*, volume 4 of *Texts in Logic and Games*, pages 151–178. Amsterdam University Press, 2008. URL: <http://www.logic.rwth-aachen.de/~ummels/knaw07.pdf>.
- 10 Ralf Küsters. Memoryless determinacy of parity games. In Erich Grädel, Wolfgang Thomas, and Thomas Wilke, editors, *Automata, Logics, and Infinite Games: A Guide to Current Research [outcome of a Dagstuhl seminar, February 2001]*, volume 2500 of *Lecture Notes in Computer Science*, pages 95–106. Springer, 2001. doi:10.1007/3-540-36387-4_6.
- 11 Noémie Meunier. *Multi-Player Quantitative Games: Equilibria and Algorithms*. PhD thesis, Université de Mons, 2016.
- 12 Christos H. Papadimitriou. *Computational complexity*. Academic Internet Publ., 2007.
- 13 Philippe Schnoebelen. The complexity of temporal logic model checking. In Philippe Balbiani, Nobu-Yuki Suzuki, Frank Wolter, and Michael Zakharyashev, editors, *Advances in Modal Logic 4, papers from the fourth conference on “Advances in Modal logic,” held in Toulouse, France, 30 September – 2 October 2002*, pages 393–436. King’s College Publications, 2002.
- 14 Michael Ummels. Rational behaviour and strategy construction in infinite multiplayer games. In S. Arun-Kumar and Naveen Garg, editors, *FSTTCS 2006: Foundations of Software Technology and Theoretical Computer Science, 26th International Conference, Kolkata, India, December 13–15, 2006, Proceedings*, volume 4337 of *Lecture Notes in Computer Science*, pages 212–223. Springer, 2006. doi:10.1007/11944836_21.
- 15 Gerd Wechsung. On the boolean closure of NP. In Lothar Budach, editor, *Fundamentals of Computation Theory, FCT ’85, Cottbus, GDR, September 9–13, 1985*, volume 199 of *Lecture Notes in Computer Science*, pages 485–493. Springer, 1985. doi:10.1007/BFb0028832.