Stackelberg-Pareto Synthesis (full version)

Véronique Bruyère
Université de Mons (UMONS), Belgium

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

Clément Tamines
Université de Mons (UMONS), Belgium

Abstract

In this paper, we study the framework of two-player Stackelberg games played on graphs in which Player 0 announces a strategy and Player 1 responds rationally with a strategy that is an optimal response. While it is usually assumed that Player 1 has a single objective, we consider here the new setting where he has several. In this context, after responding with his strategy, Player 1 gets a payoff in the form of a vector of Booleans corresponding to his satisfied objectives. Rationality of Player 1 is encoded by the fact that his response must produce a Pareto-optimal payoff given the strategy of Player 0. We study the Stackelberg-Pareto Synthesis problem which asks whether Player 0 can announce a strategy which satisfies his objective, whatever the rational response of Player 1. For games in which objectives are either all parity or all reachability objectives, we show that this problem is fixed-parameter tractable and NEXPTIME-complete. This problem is already NP-complete in the simple case of reachability objectives and graphs that are trees.

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 Stackelberg non-zero sum games played on graphs, synthesis, parity objectives

Funding This work is partially supported by the PDR project Subgame perfection in graph games (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

Two-player zero-sum infinite-duration games played on graphs are a mathematical model used to formalize several important problems in computer science, such as *reactive system synthesis*. In this context, see e.g. [27], the graph represents the possible interactions between the system and the environment in which it operates. One player models the system to synthesize, and the other player models the (uncontrollable) environment. In this classical setting, the objectives of the two players are opposite, that is, the environment is *adversarial*. Modelling the environment as fully adversarial is usually a *bold abstraction* of reality as it can be composed of one or several components, each of them having their own objective.

In this paper, we consider the framework of *Stackelberg games* [32], a richer non-zero-sum setting, in which Player 0 (the system) called *leader* announces his strategy and then Player 1 (the environment) called *follower* plays rationally by using a strategy that is an optimal response to the leader's strategy. The goal of the leader is therefore to announce a strategy that guarantees him a payoff at least equal to some given threshold. In the specific case of Boolean objectives, the leader wants to see his objective being satisfied. The concept of leader and follower is also present in the framework of *rational synthesis* [19, 25] with the difference that this framework considers several followers, each of them with their own Boolean objective. In that case, rationality of the followers is modeled by assuming that the environment settles to an equilibrium (e.g. a Nash equilibrium) where each component (composing the environment) is considered to be an *independent selfish individual*, excluding cooperation scenarios between components or the possibility of coordinated rational multiple deviations. Our work proposes a novel and natural *alternative* in which the single follower, modeling the environment, has several objectives that he wants to satisfy. After responding to the leader with his own strategy, Player 1 receives a vector of Booleans which is his payoff in the corresponding outcome. Rationality of Player 1 is encoded by the fact that he only responds in such a way to receive *Pareto-optimal payoffs*, given the strategy announced by the leader. This setting encompasses scenarios where, for instance, several components can collaborate and agree on trade-offs. The goal of the leader is therefore to announce a strategy that guarantees him to satisfy his own objective, whatever the response of the follower which ensures him a Pareto-optimal payoff. The problem of deciding whether the leader has such a strategy is called the *Stackelberg-Pareto Synthesis problem* (SPS problem).

Contributions. In addition to the definition of the new setting, our main contributions are the following ones. We consider the general class of ω-regular objectives modelled by *parity* conditions and also consider the case of *reachability* objectives for their simplicity[1]. We provide a thorough analysis of the complexity of solving the SPS problem for both objectives. Our results are interesting and singular both from a theoretical and practical point of view.

First, we show that the SPS problem is *fixed-parameter tractable* (FPT) for reachability objectives when the number of objectives of the follower is a parameter (Theorem 3) and for parity objectives when, in addition, the maximal priority used in each priority function is also a parameter of the complexity analysis (Theorem 4). These are important results as it is expected that, in practice, the *number* of objectives of the environment is limited to a few. To obtain these results, we develop a reduction from our non-zero-sum games to a zero-sum game in which the protagonist, called *Prover*, tries to show the existence of a solution to the problem, while the antagonist, called *Challenger*, tries to disprove it. We were unable to obtain FPT results with the classical tree automata techniques that are usually used in the literature for related problems, see e.g. [13, 29].

Second, we prove that the SPS problem is NEXPTIME-complete for both reachability and parity objectives (Theorems 9, 17 and 18), and that it is already NP-complete in the simple setting of reachability objectives and graphs that are trees (Theorem 15). To the best of our knowledge, this is the first NEXPTIME-completeness result for a natural class of games played on graphs. To obtain the hardness for NEXPTIME, we present a natural *succinct version* of the set cover problem that is complete for this class (Theorem 20), a result of potential independent interest. We then show how to reduce this problem to the SPS problem. Concerning the NEXPTIME-membership of the SPS problem, unfortunately, the zero-sum game used for our FPT results cannot be used directly. To obtain this result, we have shown that exponential size solutions exist for positive instances of the SPS problem and this allows us to design a nondeterministic exponential-time algorithm.

Related Work. Rational synthesis is introduced in [19] for ω-regular objectives in a setting where the followers are cooperative with the leader, and later in [25] where they are adversarial. Precise complexity results for various ω-regular objectives are established in [13] for both settings. Those complexities differ from the ones of the problem studied in this paper.

[1] Indeed, in the classical context of two-player zero-sum games, solving reachability games is in P whereas solving parity games is only known to be in NP $\cap$ co-NP, see e.g. [20].

Indeed, for reachability objectives, adversarial rational synthesis is PSPACE-complete, while for parity objectives, its precise complexity is not settled (the problem is PSPACE-hard and in NEXPTIME). Extension to non-Boolean payoffs, like mean-payoff or discounted sum, is studied in [21, 22] in the cooperative setting and in [1, 18] in the adversarial setting.

When several players (like the followers) play with the aim to satisfy their objectives, several solution concepts exist such as Nash equilibrium [26], subgame perfect equilibrium [28], secure equilibria [11, 12], or admissibility [2, 5]. The constrained existence problem, close to the cooperative rational synthesis problem, is to decide whether there exists a solution concept such that the payoff obtained by each player is larger than some threshold. Let us mention [13, 30, 31] for results on the constrained existence for Nash equilibria and [6, 7, 29] for such results for subgame perfect equilibria. The interested reader can find more pointers to works on non-zero-sum games for reactive synthesis in [4, 8].

Structure. The paper is structured as follows. In Section 2, we introduce the class of Stackelberg-Pareto games and the SPS problem. We prove in Section 3 that the SPS problem is in FPT for reachability and parity objectives. The complexity class of this problem is studied in Section 4 where we prove that it is NEXPTIME-complete and NP-complete in case of reachability objectives and graphs that are trees. In Section 5, we provide a conclusion and discuss future work.

2 Preliminaries and Stackelberg-Pareto Synthesis Problem

This section introduces the class of two-player Stackelberg-Pareto games in which the first player has a single objective and the second has several. We present a decision problem on those games called the Stackelberg-Pareto Synthesis problem, which we study in this paper.

2.1 Preliminaries

Game Arena. 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 call a game arena a *tree arena* if it is a tree in which every leaf vertex has itself as its only successor. 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.

Plays. A *play* in a game arena G is an infinite sequence of vertices $\rho = v_0 v_1 \ldots \in V^\omega$ 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 sequences $h = v_0 \ldots v_j \in V^+$ defined similarly. A history is *elementary* if it contains no cycles. We denote by Plays_G the set of plays in G. We write Hist_G (resp. $\mathsf{Hist}_{G,i}$) the set of histories (resp. histories ending with a vertex in V_i). We use the notations Plays, Hist, and Hist_i when G is clear from the context. We write $\mathsf{Occ}(\rho)$ the set of vertices occurring in ρ and $\mathsf{Inf}(\rho)$ the set of vertices occurring infinitely often in ρ.

Strategies. 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 $(v, v') \in E$. It is *memoryless* if $\sigma_i(hv) = \sigma_i(h'v)$ for all histories $hv, h'v$ ending with the same vertex $v \in V_i$. More generally, it is *finite-memory* if it can be encoded by a Moore machine $\mathcal{M}$ [20]. The *memory size* of σ_i is the number of memory states of $\mathcal{M}$. In particular, σ_i is memoryless when it has a memory size of one.

Given a strategy σ_i of Player i, 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. We denote by $\mathsf{Plays}_{\sigma_i}$ (resp. Hist_{σ_i}) the set of plays (resp. histories) consistent with σ_i. A *strategy profile* is a tuple $\sigma = (\sigma_0, \sigma_1)$ of strategies, one for each player. We write $\mathsf{out}(\sigma)$ the unique play consistent with both strategies and we call it the *outcome* of σ.

Objectives. An *objective* for Player i is a set of plays $\Omega \subseteq \mathsf{Plays}$. A play ρ *satisfies* the objective Ω if $\rho \in \Omega$. In this paper, we focus on the two following ω-regular objectives. Let $T \subseteq V$ be a subset of vertices called a *target set*, the *reachability* objective $\mathsf{Reach}(T) = \{\rho \in \mathsf{Plays} \mid \mathsf{Occ}(\rho) \cap T \neq \emptyset\}$ asks to visit at least one vertex of T. Let $c : V \to \mathbb{N}$ be a function called a *priority function* which assigns an integer to each vertex in the arena, the *parity* objective $\mathsf{Parity}(c) = \{\rho \in \mathsf{Plays} \mid \min_{v \in \mathsf{Inf}(\rho)}(c(v)) \text{ is even}\}$ asks that the minimum priority visited infinitely often be even.

2.2 Stackelberg-Pareto Synthesis Problem

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. In this paper, we focus on SP games where the objectives are either all reachability or all parity objectives and call such games *reachability* (resp. *parity*) *SP games*.

Payoffs in SP Games. The *payoff* of a play $\rho \in \mathsf{Plays}$ corresponds to 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. Note that we omit to include 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 $(\mathsf{won}(\rho), \mathsf{pay}(\rho))$ the *extended payoff* of ρ. Given a strategy profile σ, we write $\mathsf{won}(\sigma) = \mathsf{won}(\mathsf{out}(\sigma))$ and $\mathsf{pay}(\sigma) = \mathsf{pay}(\mathsf{out}(\sigma))$. For reachability SP games, since reachability objectives are prefix-dependant and given a history $h \in \mathsf{Hist}$, we also define $\mathsf{won}(h)$ and $\mathsf{pay}(h)$ as done for plays.

We introduce 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 $p < p'$. A subset of payoffs $P \subseteq \{0,1\}^t$ is an *antichain* if it is composed of pairwise incomparable payoffs with respect to $\leq$.

Stackelberg-Pareto Synthesis Problem. Given a strategy σ_0 of Player 0, we 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 it is an antichain. We say that those payoffs are σ_0-*fixed Pareto-optimal* and write $|P_{\sigma_0}|$ the number of such payoffs. A play $\rho \in \mathsf{Plays}_{\sigma_0}$ is called σ_0-fixed Pareto-optimal if its payoff $\mathsf{pay}(\rho)$ is in P_{σ_0}.

The problem studied in this paper asks whether there exists a strategy σ_0 for Player 0 such that every play in $\mathsf{Plays}_{\sigma_0}$ which is σ_0-fixed Pareto-optimal satisfies the objective of Player 0. This corresponds to the assumption that given a strategy of Player 0, Player 1 will play *rationally*, that is, with a strategy σ_1 such that $\mathsf{out}((\sigma_0, \sigma_1))$ is σ_0-fixed Pareto-optimal. It is therefore sound to ask that Player 0 wins against such rational strategies.

▶ **Definition 1.** *Given an SP game, the* Stackelberg-Pareto Synthesis problem *(SPS problem) is to decide whether there exists a strategy σ_0 for Player 0 (called a* solution*) such that for each strategy profile $\sigma = (\sigma_0, \sigma_1)$ with $\mathsf{pay}(\sigma) \in P_{\sigma_0}$, it holds that $\mathsf{won}(\sigma) = 1$.*

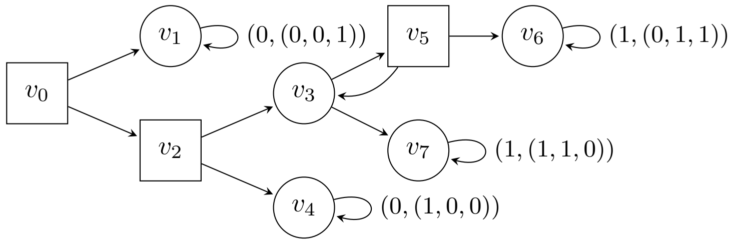

Figure 1 A reachability SP game.

Witnesses. Given a strategy σ_0 that is a solution to the SPS problem and any payoff $p \in P_{\sigma_0}$, for each play ρ consistent with σ_0 such that $\mathsf{pay}(\rho) = p$ it holds that $\mathsf{won}(\rho) = 1$. For each $p \in P_{\sigma_0}$, we arbitrarily select such a play which we call a *witness* (of p). We denote by Wit_{σ_0} the set of all witnesses, of which there are as many as payoffs in P_{σ_0}. In the sequel, it is useful to see this set as a tree composed of $|\mathsf{Wit}_{\sigma_0}|$ branches. Additionally for a given history $h \in \mathsf{Hist}$, we write $\mathsf{Wit}_{\sigma_0}(h)$ the set of witnesses for which h is a prefix, i.e., $\mathsf{Wit}_{\sigma_0}(h) = \{\rho \in \mathsf{Wit}_{\sigma_0} \mid h \text{ is prefix of } \rho\}$. Notice that $\mathsf{Wit}_{\sigma_0}(h) = \mathsf{Wit}_{\sigma_0}$ when $h = v_0$ and that $\mathsf{Wit}_{\sigma_0}(h)$ decreases as h increases, until it contains a single value or becomes empty.

▶ **Example 2.** Consider the reachability SP game with arena G depicted in Figure 1 in which Player 1 has $t = 3$ objectives. The vertices of Player 0 (resp. Player 1) are depicted as circles (resp. squares)[2]. Every objective in the game is a reachability objective defined as follows: $\Omega_0 = \mathsf{Reach}(\{v_6, v_7\})$, $\Omega_1 = \mathsf{Reach}(\{v_4, v_7\})$, $\Omega_2 = \mathsf{Reach}(\{v_3\})$, $\Omega_3 = \mathsf{Reach}(\{v_1, v_6\})$. The extended payoff of plays reaching vertices from which they can only loop is displayed in the arena next to those vertices, and the extended payoff of play $v_0v_2(v_3v_5)^\omega$ is $(0,(0,1,0))$.

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 $\rho = v_0v_2(v_4)^\omega$ is consistent with σ_0, has payoff $(1,0,0)$ and is lost by Player 0. Strategy σ_0 is therefore not a solution to the SPS problem. In this game, there is only one other memoryless strategy for Player 0, where he chooses to always move to v_7 from v_3. One can verify that it is again not a solution to the SPS problem.

We can however define a finite-memory strategy σ'_0 such that $\sigma'_0(v_0v_2v_3) = v_5$ and $\sigma'_0(v_0v_2v_3v_5v_3) = v_7$ and show that it is a solution to the problem. Indeed, the set of σ'_0-fixed 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 set $\mathsf{Wit}_{\sigma'_0}$ of witnesses for these payoffs is $\{v_0v_2v_3v_5v_6^\omega, v_0v_2v_3v_5v_3v_7^\omega\}$ and is in this case the unique set of witnesses. This example shows that Player 0 sometimes needs memory in order to have a solution to the SPS problem.

3 Fixed-Parameter Complexity

In this section, we show that the SPS problem is in FPT for both cases of reachability and parity SP games. The details of our proof for each type of objective are provided separately in their own subsection. We refer the reader to [15] for the concept of fixed-parameter complexity.

[2] This convention is used throughout this paper.

▶ **Theorem 3.** *Solving the SPS problem is in* FPT *for reachability SP games for parameter* t *equal to the number of objectives of Player* 1*.*

▶ **Theorem 4.** *Solving the SPS problem is in* FPT *for parity SP games for parameters* t *and the maximal priority according to each parity objective of Player* 1*.*

3.1 Challenger-Prover Game

In order to prove Theorem 3 and Theorem 4, we provide a reduction to a specific two-player zero-sum game, called the *Challenger-Prover* game (C-P game). This game is a *zero-sum*[3] game played between *Challenger* (written $\mathcal{C}$) and *Prover* (written $\mathcal{P}$). We will show that Player 0 has a solution to the SPS problem in an SP game if and only if $\mathcal{P}$ has a winning strategy in the corresponding C-P game. In the latter game, $\mathcal{P}$ tries to show the existence of a strategy σ_0 that is solution to the SPS problem in the original game and $\mathcal{C}$ tries to disprove it. The C-P game is described independently of the objectives used in the SP game and its objective is described as such in a *generic way*. We later provide the proof of our FPT results by adapting it specifically for reachability and parity SP games.

Intuition on the C-P Game. Without loss of generality, the SP games we consider in this section are such that each vertex in their arena has at most *two successors*. It can be shown (see Appendix A) that any SP game $\mathcal{G}$ with n vertices can be transformed into an SP game $\bar{\mathcal{G}}$ with $\mathcal{O}(n^2)$ vertices such that every vertex has at most two successors and Player 0 has a solution to the SPS problem in $\mathcal{G}$ if and only if he has a solution to the SPS problem in $\bar{\mathcal{G}}$.

Let $\mathcal{G}$ be an SP game. The C-P game $\mathcal{G}'$ is a zero-sum game associated with $\mathcal{G}$ that intuitively works as follows. First, $\mathcal{P}$ selects a set P of payoffs which he announces as the set of Pareto-optimal payoffs P_{σ_0} for the solution σ_0 to the SPS problem in $\mathcal{G}$ he is trying to construct. Then, $\mathcal{P}$ tries to show that there exists a set of witnesses Wit_{σ_0} in $\mathcal{G}$ for the payoffs in P. After the selection of P in $\mathcal{G}'$, there is a one-to-one correspondence between plays in the arenas G and G' such that the vertices in G' are augmented with a set W which is a subset of P. Initially W is equal to P and after some history in G', W contains payoff p if the corresponding history in G is prefix of the witness with payoff p in the set Wit_{σ_0} that $\mathcal{P}$ is building. In addition, the objective $\Omega_{\mathcal{P}}$ of $\mathcal{P}$ is such that he has a winning strategy $\sigma_{\mathcal{P}}$ in $\mathcal{G}'$ if and only if the set P that he selected coincides with the set P_{σ_0} for the corresponding strategy σ_0 in $\mathcal{G}$ and the latter strategy is a solution to the SPS problem in $\mathcal{G}$. A part of the arena of the C-P game for Example 2 with a highlighted winning strategy for $\mathcal{P}$ is illustrated in Figure 2.

Arena of the C-P Game. The initial vertex $\perp$ belongs to $\mathcal{P}$. From this vertex, he selects a successor (v_0, P, W) such that $W = P$ and P is an antichain of payoffs which $\mathcal{P}$ announces as the set P_{σ_0} for the strategy σ_0 in G he is trying to construct. All vertices in plays starting with this vertex will have this same value for their P-component. Those vertices are either a triplet (v, P, W) that belongs to $\mathcal{P}$ or $(v, P, (W_l, W_r))$ that belongs to $\mathcal{C}$. Given a play ρ (resp. history h) in G', we denote by ρ_V (resp. h_V) the play (resp. history) in G obtained by removing $\perp$ and keeping the v-component of every vertex of $\mathcal{P}$ in ρ (resp. h), which we call its *projection*.

[3] We suppose the reader familiar with the concept of zero-sum games, see e.g. [20].

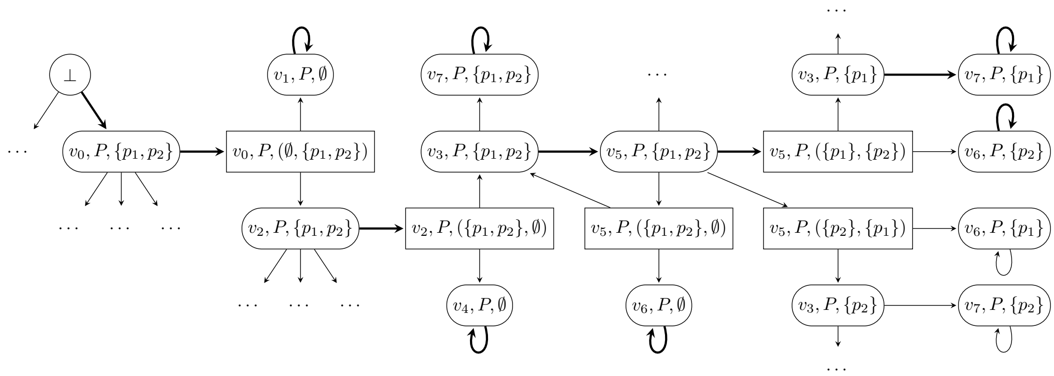

Figure 2 A part of the C-P game for Example 2 with $P = \{p_1, p_2\}, p_1 = (1, 1, 0)$ and $p_2 = (0, 1, 1)$.

- After history hm such that $m = (v, P, W)$ with $v \in V_0$, $\mathcal{P}$ selects a successor v' such that $(v, v') \in E$ and vertex (v', P, W) is added to the play. This corresponds to Player 0 choosing a successor v' after history $h_V v$ in G.
- After history hm such that $m = (v, P, W)$ with $v \in V_1$, $\mathcal{P}$ selects a successor $(v, P, (W_l, W_r))$ with (W_l, W_r) a partition of W. This corresponds to $\mathcal{P}$ splitting the set W into two parts according to the two successors v_l and v_r of v. For the strategy σ_0 that $\mathcal{P}$ tries to construct and its set of witnesses Wit_{σ_0} he is building, he asserts that W_l (resp. W_r) is the set of payoffs of the witnesses in $\mathsf{Wit}_{\sigma_0}(h_V v_l)$ (resp. $\mathsf{Wit}_{\sigma_0}(h_V v_r)$).
- From a vertex $(v, P, (W_l, W_r))$, $\mathcal{C}$ can select a successor (v_l, P, W_l) or (v_r, P, W_r) which corresponds to the choice of Player 1.

Formally, the game arena of the C-P game is the tuple $G' = (V', V'_{\mathcal{P}}, V'_{\mathcal{C}}, E', \perp)$ with
- $V'_{\mathcal{P}} = \{\perp\} \cup \{(v, P, W) \mid v \in V, P \subseteq \{0,1\}^t$ is an antichain and $W \subseteq P\}$,
- $V'_{\mathcal{C}} = \{(v, P, (W_l, W_r)) \mid v \in V_1, P \subseteq \{0,1\}^t$ is an antichain and $W_l, W_r \subseteq P\}$,
- $(\perp, (v, P, W)) \in E'$ if $v = v_0$ and $P = W$,
- $((v, P, W), (v', P, W)) \in E'$ if $v \in V_0$ and $(v, v') \in E$,
- $((v, P, W), (v, P, (W_l, W_r))) \in E'$ if $v \in V_1$ and (W_l, W_r) is a partition of W,
- $((v, P, (W_l, W_r)), (v', P, W)) \in E'$ if $v' = v_l$ and $W = W_l$ or $v' = v_r$ and $W = W_r$.

In the definition of E', if v has a single successor v' in G, it is assumed to be v_l and W_r is always equal to $\emptyset$. We use as a convention that given the two successors v_i and v_j of vertex v, v_i is the left successor if $i < j$.

Objective of $\mathcal{P}$ in the C-P Game. Let us now discuss the objective $\Omega_{\mathcal{P}}$ of $\mathcal{P}$. The W-component of the vertices controlled by $\mathcal{P}$ has a size that decreases along a play ρ in G'. We write $lim_W(\rho)$ the value of the W-component at the limit in ρ. Recall that with this W-component, $\mathcal{P}$ tries to construct a solution σ_0 to the SPS problem with associated sets P_{σ_0} and Wit_{σ_0}. Therefore, for him to win in the C-P game, $lim_W(\rho)$ must be a singleton or empty in every consistent play such that:

- $lim_W(\rho)$ must be a singleton $\{p\}$ with p the payoff of ρ_V in G, showing that $\rho_V \in \mathsf{Wit}_{\sigma_0}$ is a correct witness for p. In addition, it must hold that $\mathsf{won}(\rho_V) = 1$ as $p \in P$ and as $\mathcal{P}$ wants σ_0 to be a solution.
- $lim_W(\rho)$ must be the empty set such that either the payoff of ρ_V belongs to P_{σ_0} and $\mathsf{won}(\rho_V) = 1$, or the payoff of ρ_V is strictly smaller than some payoff in P_{σ_0}.

These three conditions verify that the sets $P = P_{\sigma_0}$ and Wit_{σ_0} are correct and that σ_0 is indeed a solution to the SPS problem in G. They are generic as they do not depend on the actual objectives used in the SP game.

Let us give the formal definition of $\Omega_\mathcal{P}$. For an antichain P of payoffs, we write $\mathsf{Plays}^P_{G'}$ the set of plays in G' which start with $\bot(v_0, P, P)$ and we define the following set

$$B_P = \big\{\rho \in \mathsf{Plays}^P_{G'} \mid (lim_W(\rho) = \{p\} \wedge \mathsf{pay}(\rho_V) = p \in P \wedge \mathsf{won}(\rho_V) = 1) \vee \tag{1}$$
$$(lim_W(\rho) = \emptyset \wedge \mathsf{pay}(\rho_V) \in P \wedge \mathsf{won}(\rho_V) = 1) \vee \tag{2}$$
$$(lim_W(\rho) = \emptyset \wedge \exists p \in P, \mathsf{pay}(\rho_V) < p)\big\}. \tag{3}$$

Objective $\Omega_\mathcal{P}$ of $\mathcal{P}$ in $\mathcal{G}'$ is the union of B_P over all antichains P. As the C-P game is zero-sum, objective $\Omega_\mathcal{C}$ equals $\mathsf{Plays}_{G'} \setminus \Omega_\mathcal{P}$. The following theorem holds.

▶ **Theorem 5.** *Player 0 has a strategy σ_0 that is solution to the SPS problem in $\mathcal{G}$ if and only if $\mathcal{P}$ has a winning strategy $\sigma_\mathcal{P}$ from $\bot$ in the C-P game $\mathcal{G}'$.*

Proof of Theorem 5. Let us first assume that Player 0 has a strategy σ_0 that is solution to the SPS problem in $\mathcal{G}$. Let P_{σ_0} be its set of σ_0-fixed Pareto-optimal payoffs and let Wit_{σ_0} be a set of witnesses. We construct the strategy $\sigma_\mathcal{P}$ from σ_0 such that

- $\sigma_\mathcal{P}(\bot) = (v_0, P, P)$ such that $P = P_{\sigma_0}$ (this vertex exists as P_{σ_0} is an antichain),
- $\sigma_\mathcal{P}(hm) = (v', P, W)$ if $m = (v, P, W)$ with $v \in V_0$ and $v' = \sigma_0(h_V v)$,
- $\sigma_\mathcal{P}(hm) = (v, P, (W_l, W_r))$ if $m = (v, P, W)$ with $v \in V_1$ and for $i \in \{l, r\}$, $W_i = \{\mathsf{pay}(\rho) \mid \rho \in \mathsf{Wit}_{\sigma_0}(h_V v_i)\}$.

It is clear that given a play ρ in G' consistent with $\sigma_\mathcal{P}$, the play ρ_V in G is consistent with σ_0. Let us show that $\sigma_\mathcal{P}$ is winning for $\mathcal{P}$ from $\bot$ in G'. Consider a play ρ in G' consistent with $\sigma_\mathcal{P}$. There are two possibilities. *(i)* ρ_V is a witness of Wit_{σ_0} and by construction $lim_W(\rho) = \{p\}$ with $p = \mathsf{pay}(\rho_V)$; thus $\mathsf{won}(\rho_V) = 1$ as σ_0 is a solution and ρ_V is a witness. *(ii)* ρ_V is not a witness and by construction $lim_W(\rho) = \emptyset$; as σ_0 is a solution, then $p = \mathsf{pay}(\rho_V)$ is bounded by some payoff of P_{σ_0} and in case of equality $\mathsf{won}(\rho_V) = 1$. Therefore ρ satisfies the objective B_P of $\Omega_\mathcal{P}$ since it satisfies condition (1) in case *(i)* and condition (2) or (3) in case *(ii)*.

Let us now assume that $\mathcal{P}$ has a winning strategy $\sigma_\mathcal{P}$ from $\bot$ in G'. Let P be the antichain of payoffs chosen from $\bot$ by this strategy. We construct the strategy σ_0 from $\sigma_\mathcal{P}$ such that $\sigma_0(h_V v) = v'$ given $\sigma_\mathcal{P}(hm) = (v', P, W)$ with $m = (v, P, W)$ and $v \in V_0$. Notice that this definition makes sense since there is a unique history hm ending with a vertex of $\mathcal{P}$ associated with $h_V v$ showing a one-to-one correspondence between those histories.

Let us show σ_0 is a solution to the SPS problem with P_{σ_0} being the set P. First notice that P is not empty. Indeed let ρ be a play consistent with $\sigma_\mathcal{P}$. As ρ belongs to $\Omega_\mathcal{P}$ and in particular to B_P, one can check that $P \neq \emptyset$ by inspecting conditions (1) to (3). Second notice that by definition of E', if $((v, P, W), (v, P, (W_l, W_r))) \in E'$ with $W \neq \emptyset$, then either W_l or W_r is not empty. Therefore given any payoff $p \in P$, there is a unique play ρ consistent with $\sigma_\mathcal{P}$ such that $lim_W(\rho) = \{p\}$. By construction of σ_0 and as $\sigma_\mathcal{P}$ is winning, the play ρ_V is consistent with σ_0, has payoff p, and is won by Player 0 (see (1)).

Let ρ_V be a play consistent with σ_0 and ρ be the corresponding play consistent with $\sigma_\mathcal{P}$. It remains to consider (2) and (3). These conditions indicate that ρ_V has a payoff equal to or strictly smaller than a payoff in P and that in case of equality $\mathsf{won}(\rho_V) = 1$. This shows that $P_{\sigma_0} = P$ and that σ_0 is a solution to the SPS problem. ◀

3.2 Fixed-Parameter Complexity of Reachability SP Games

We now develop the proof of Theorem 3 which works by specializing the generic objective $\Omega_\mathcal{P}$ to handle reachability SP games. We extend the arena G' of the C-P game such that its vertices keep track of the objectives of $\mathcal{G}$ which are satisfied along a play. Given an extended payoff $(w, p) \in \{0,1\} \times \{0,1\}^t$ and a vertex $v \in V$, we define the *payoff update* $\mathsf{upd}(w, p, v) = (w', p')$ such that

$$\begin{aligned} w' = 1 \quad &\iff \quad w = 1 \text{ or } v \in T_0, \\ p'_i = 1 \quad &\iff \quad p_i = 1 \text{ or } v \in T_i, \quad \forall i \in \{1, \dots, t\}. \end{aligned}$$

We obtain the extended arena G^* as follows: *(i)* its set of vertices is $V' \times \{0,1\} \times \{0,1\}^t$, *(ii)* its initial vertex is $\bot^* = (\bot, 0, (0, \dots, 0))$, and *(iii)* $((m, w, p), (m', w', p'))$ with $m' = (v', P, W)$ or $m' = (v', P, (W_l, W_r))$ is an edge in G^* if $(m, m') \in E'$ and $(w', p') = \mathsf{upd}(w, p, v')$.

We define the zero-sum game $\mathcal{G}^* = (G^*, \Omega^*_{\mathcal{P}})$ in which the three abstract conditions (1-3) detailed previously are encoded into the following Büchi objective by using the (w, p)-component added to vertices. We define $\Omega^*_{\mathcal{P}} = \mathsf{Büchi}(B^*)$ with

$$\begin{aligned} B^* = \big\{(v, P, W, w, p) \in V^*_{\mathcal{P}} \;\mid\; &(W = \{p\} \wedge w = 1) \vee && (1') \\ &(W = \emptyset \quad \wedge p \in P \wedge w = 1) \vee && (2') \\ &(W = \emptyset \quad \wedge \exists p' \in P,\ p < p')\big\}. && (3') \end{aligned}$$

▶ **Proposition 6.** *Player 0 has a strategy σ_0 that is solution to the SPS problem in a reachability SP game $\mathcal{G}$ if and only if $\mathcal{P}$ has a winning strategy $\sigma^*_{\mathcal{P}}$ in $\mathcal{G}^*$.*

The proof of this proposition is a consequence of Theorem 5. Using the one-to-one correspondence between plays in G and plays in G^* and the fact that $\mathcal{G}$ is a reachability SP game, the (w, p)-component in vertices of G^* allows us to easily retrieve the extended payoff of a play in G. Indeed, in a play $\rho \in \mathsf{Plays}_{G^*}$, given the construction of G^* and the payoff update function, it holds that from some point on the W- and (w, p)-components are constant. Therefore it holds that $w = \mathsf{won}(\rho_V)$, $p = \mathsf{pay}(\rho_V)$ and $W = lim_W(\rho)$ for that play ρ. Moreover the P-component is constant along a play in G^*. It is direct to see that the plays ρ in G^* which visit infinitely often the set B^*, and therefore satisfy the Büchi objective $\Omega^*_{\mathcal{P}} = \mathsf{Büchi}(B^*)$, satisfy one of the three conditions (1-3) stated in Subsection 3.1. The converse is also true.

We now describe a FPT algorithm for deciding the existence of a solution to the SPS problem in a reachability SP game, thus proving Theorem 3.

Proof of Theorem 3. We describe the following FPT algorithm (for parameter t) for deciding the existence of a solution to the SPS problem in a reachability SP game $\mathcal{G}$ by using Proposition 6. First, we construct the zero-sum game $\mathcal{G}^*$. Its number n of vertices is upper-bounded by $1 + |V| \cdot 2^{2^{t+1}} \cdot 2^{t+1} + |V| \cdot 2^{3 \cdot 2^t} \cdot 2^{t+1}$. Indeed, except the initial vertex, vertices are of the form either (v, P, W, w, p) or $(v, P, (W_l, W_r), w, p)$ such that P, W, W_l and W_r are antichains of payoffs in $\{0,1\}^t$, and (w, p) is an extended payoff. The construction of $\mathcal{G}^*$ is thus in FPT for parameter t. Second, By Proposition 6, deciding whether there exists a solution to the SPS problem in $\mathcal{G}$ amounts to deciding if $\mathcal{P}$ has a winning strategy from $\bot^*$ in $\mathcal{G}^*$. Since the objective $\Omega^*_{\mathcal{P}}$ of $\mathcal{P}$ in $\mathcal{G}^*$ is a Büchi objective, this game can be solved in $\mathcal{O}(n^2)$ [10]. It follows that $\mathcal{G}^*$ can be solved in FPT for parameter t. ◀

3.3 Fixed-Parameter Complexity of Parity SP Games

We now turn to parity SP games and explain why solving the SPS problem in these games is in FPT, again by reduction to the C-P game. To this end, we first recall the notion of Boolean Büchi games.

Boolean Büchi games are zero-sum games which we use in our reduction to the C-P for parity SP game. 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* 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 sets to be visited infinitely often by a play is satisfied. We denote by $|\phi|$ the size of ϕ as equal to the number of conjunctions and disjunctions in ϕ. The following theorem on the fixed-parameter complexity of Boolean Büchi games is proved in [9].

▶ **Theorem 7.** *Solving Boolean Büchi games is in* FPT*, with an algorithm in* $\mathcal{O}(2^M \cdot |\phi| + (M^M \cdot |V|)^5)$ *time with* $M = 2^m$ *such that* m *is the number of variables and* $|\phi|$ *is the size of* ϕ *in the Boolean Büchi objective [9].*

Let $\mathcal{G} = (G, \Omega_0, \dots, \Omega_t)$ be a parity SP game with parity objectives such that $\Omega_i = \mathsf{Parity}(c_i)$ for a priority function $c_i : V \to \mathbb{N}$. Let G' be the arena of the C-P game. In the following, we construct a Boolean Büchi objective $\Omega'_{\mathcal{P}}$ for $\mathcal{P}$ such that the following proposition holds.

▶ **Proposition 8.** *Player* 0 *has a strategy* σ_0 *that is solution to the SPS problem in* $\mathcal{G}$ *if and only if* $\mathcal{P}$ *has a winning strategy* $\sigma_{\mathcal{P}}$ *in* $\mathcal{G}^* = (G', \Omega'_{\mathcal{P}})$.

Proof of Proposition 8. let $G' = (V', V'_{\mathcal{P}}, V'_{\mathcal{C}}, E', \perp)$ be the arena of the C-P game presented in Section 3.1. Recall that the objective $\Omega_{\mathcal{P}}$ of this game is the union of the sets B_P over all antichains P such that B_P is the disjunction of conditions (1-3). The idea of the proof is to translate this objective into a Boolean Büchi objective $\Omega'_{\mathcal{P}}$. We will proceed step by step. The required Boolean formula for defining $\Omega'_{\mathcal{P}}$ is equal to

$$\phi = \bigvee_{P} \left(x_P \wedge (cond_1^P \vee cond_2^P \vee cond_3^P) \right)$$

such that the main disjunction is over all antichains P. The variable x_P corresponds to the set $T_P = \{(v, P, W) \in V'_{\mathcal{P}}\}$. The valuation of x_P is true for a given play if and only if the set T_P is visited infinitely often and therefore P is the antichain chosen by $\mathcal{P}$ in G'. Since the P-component is constant along a play, only one x_P is valued as true for a given play. Let us now detail each subformula $cond_i^P$ that is the translation of condition (i) for $i \in \{1, 2, 3\}$.

Let us begin with the encoding of payoffs. Let $d_0, \dots, d_t$ be such that d_i is the maximal even priority appearing in G according to priority function c_i for objective Ω_i with $i \in \{0, \dots, t\}$.

First, we show that a parity objective $\mathsf{Parity}(c_i)$ from $\mathcal{G}$ can be encoded as a Boolean Büchi objective. Given the parity objective $\mathsf{Parity}(c_i)$, we construct the Boolean formula $parity_i$ over variables $\{x_0^i, x_1^i, \dots, x_{d_i}^i\}$ such that

$$parity_i = x_0^i \vee (x_2^i \wedge \neg x_1^i) \vee \dots \vee (x_{d_i}^i \wedge \neg x_{d_i - 1}^i \wedge \neg x_{d_i - 3}^i \wedge \dots \wedge \neg x_1^i)$$

and for $j \in \{0, \dots, d_i\}$, the set corresponding to variable x_j^i is $T_j^i = \{(v, P, W) \in V'_{\mathcal{P}} \mid c_i(v) = j\}$. It is easy to show that the parity objective $\mathsf{Parity}(c_i)$ is satisfied if and only if the Boolean Büchi objective $\mathsf{BooleanBüchi}(parity_i, T_0^i, \dots, T_{d_i}^i)$ is satisfied.

Second, given a payoff $p = (p_1, \dots, p_t)$ in $\mathcal{G}$, we consider the Boolean formula

$$payoff_p = C_1 \wedge \dots \wedge C_t$$

such that $C_i = \mathit{parity}_i$ if $p_i = 1$ and $C_i = \neg\mathit{parity}_i$ otherwise. Clearly the projection ρ_V of play ρ realizes payoff p if and only if ρ satisfies the Boolean Büchi objective $\mathsf{BooleanBüchi}(\mathit{payoff}_p, T^1_0, \dots, T^1_{d_1}, \dots, T^t_0, \dots, T^t_{d_t})$.

We now fix some antichain P. Let us detail subformula cond^P_1 encoding condition (1). For a payoff p, we define formula

$$\mathit{single}_p = x_p \wedge \mathit{payoff}_p \wedge \mathit{parity}_0$$

such that $x_p = \{(v, P, W) \in V'_{\mathcal{P}} \mid W = \{p\}\}$. Since at some point during a play, the W-component stabilizes and since a play ρ which satisfies $\mathit{payoff}_p \wedge \mathit{parity}_0$ is such that $\mathsf{pay}(\rho_V) = p$ and $\mathsf{won}(\rho_V) = 1$, it holds that satisfying this formula corresponds exactly to satisfying condition (1) for some p. Formula cond^P_1 is thus the disjunction

$$\mathit{cond}^P_1 = \bigvee_{p \in P} \mathit{single}_p.$$

Let us shift to subformula cond^P_2 encoding condition (2). Using similar arguments, we define for payoff p formula

$$\mathit{empty}_p = x_\emptyset \wedge \mathit{payoff}_p \wedge \mathit{parity}_0$$

such that $x_\emptyset = \{(v, P, \emptyset) \in V'_{\mathcal{P}}\}$. It corresponds exactly to the set of plays ρ such that $\mathit{lim}_W(\rho) = \emptyset$, $\mathsf{pay}(\rho_V) = p$ and $\mathsf{won}(\rho_V) = 1$. Therefore

$$\mathit{cond}^P_2 = \bigvee_{p \in P} \mathit{empty}_p.$$

Finally, we define subformula cond^P_3 encoding condition (3). Let $\overline{P}$ be the set containing every payoff p' such that $\exists p \in P,\ p' < p$. We define

$$\mathit{cond}^P_3 = \bigvee_{p' \in \overline{P}} \mathit{smaller}_{p'}$$

with $\mathit{smaller}_{p'}$ being the formula $x_\emptyset \wedge \mathit{payoff}_{p'}$.

Notice that the Boolean formula ϕ constructed in this proof has a number m of variables and a size $|\phi|$ that only depend on t and d_i, $i \in \{0, \dots, t\}$. ◀

This previous construction can be used to provide the proof of Theorem 4.

Proof of Theorem 4. We describe the following FPT algorithm for deciding the existence of a solution to the SPS problem in a parity SP game $\mathcal{G}$ by using Proposition 8. First, we construct the zero-sum game $\mathcal{G}^*$ of Proposition 8. Its number n of vertices is upper-bounded by $1 + |V| \cdot 2^{2^{t+1}} + |V| \cdot 2^{3 \cdot 2^t}$ which is in $\mathcal{O}(|V| \cdot f(t))$ with f a computable function which only depends on t. Moreover the number m of variables and the size $|\phi|$ of the Boolean formula ϕ defining the Boolean Büchi objective of $\mathcal{G}^*$ depend only on parameters t and d_i for $i \in \{0, \dots, t\}$. Therefore the construction of $\mathcal{G}^*$ is in FPT for these parameters. Deciding whether there exists a solution to the SPS problem in $\mathcal{G}$ amounts to deciding if $\mathcal{P}$ has a winning strategy from $\perp$ in $\mathcal{G}^*$. By Theorem 7, the latter Boolean Büchi game can be solved with an algorithm in $\mathcal{O}(2^M \cdot |\phi| + (M^M \cdot n)^5)$ time with $M = 2^m$. It follows that $\mathcal{G}^*$ can be solved in $\mathcal{O}(2^M \cdot |\phi| + (M^M \cdot |V| \cdot f(t))^5)$ which is in FPT for the announced parameters. ◀

4 Complexity Class of the SPS Problem

In this section, we study the complexity class of the SPS problem and prove its $\mathsf{NEXPTIME}$-completeness for both reachability and parity SP games.

4.1 NEXPTIME-Membership

We first show the membership to NEXPTIME of the SPS problem by providing a nondeterministic algorithm with time exponential in the size of the game $\mathcal{G}$. By *size*, we mean the number $|V|$ of its vertices and the number t of objectives of Player 1.

▶ **Theorem 9.** *The SPS problem is in* NEXPTIME *for reachability and parity SP games.*

We show that the SPS problem is in NEXPTIME by proving that if Player 0 has a strategy which is a solution to the problem, then he has one which is finite-memory with at most an exponential number of memory states[4]. This yields a NEXPTIME algorithm in which we nondeterministically guess such a strategy and check in exponential time that it is indeed a solution to the problem.

▶ **Proposition 10.** *Let $\mathcal{G}$ be a reachability SP game or a parity SP game. Let σ_0 be a solution the the SPS problem. Then there exists another solution $\tilde{\sigma}_0$ that is finite-memory and has a memory size exponential in the size of $\mathcal{G}$.*

While the proof of Proposition 10 requires some specific arguments to treat both reachability and parity objectives, it is based on the following common principles.

- We start from a winning strategy σ_0 for the SPS problem and the objectives $\Omega_0, \Omega_1, \ldots, \Omega_t$ and consider a set of witnesses Wit_{σ_0}, that contains one play for each element of the set P_{σ_0} of σ_0-fixed Pareto-optimal payoffs.
- We start by showing the existence of a strategy $\hat{\sigma}_0$ constructed from σ_0, in which Player 0 follows σ_0 as long as the current consistent history is prefix of at least one witness in Wit_{σ_0}. Then when a deviation from Wit_{σ_0} occurs, Player 0 switches to a *punishing strategy*. A deviation is a history that leaves the set of witnesses Wit_{σ_0} after a move of Player 1 (this is not possible by a move of Player 0). After such a deviation, $\hat{\sigma}_0$ systematically imposes that the consistent play either satisfies Ω_0 or is not σ_0-fixed Pareto-optimal, i.e., it gives to Player 1 a payoff that is strictly smaller than the payoff of a witness in Wit_{σ_0}. This makes the deviation *irrational* for Player 1. We show that this can be done, both for reachability and parity objectives, with at most exponentially many different punishing strategies, each having a size bounded exponentially in the size of the game. The strategy $\hat{\sigma}_0$ that we obtain is therefore composed of the part of σ_0 that produces Wit_{σ_0} and a punishment part whose size is at most exponential.
- Then, we show how to decompose each witness in Wit_{σ_0} into at most exponentially many *sections* that can, in turn, be compacted into finite elementary paths or lasso shaped paths of polynomial length. As Wit_{σ_0} contains exactly $|P_{\sigma_0}|$ witnesses ρ, those compact witnesses $c\rho$ can be produced by a finite-memory strategy with an exponential size for both reachability and parity objectives. This allows us to construct a strategy $\tilde{\sigma}_0$ that produces the compact witnesses and acts as $\hat{\sigma}_0$ after any deviation. This strategy is a solution of the SPS problem and has an exponential size as announced.

We now develop the details of the construction of the strategies $\hat{\sigma}_0$ and $\tilde{\sigma}_0$. Figure 3 illustrates this construction. It is done in several steps to finally get the proof of Proposition 10. For the rest of this section, we fix an SP game $\mathcal{G}$ with objectives $\Omega_0, \Omega_1, \ldots, \Omega_t$, a strategy σ_0 that is solution to the SPS problem, a set of witnesses Wit_{σ_0} for the σ_0-fixed Pareto-optimal payoffs in P_{σ_0}, and we write $\Omega^{<}(P_{\sigma_0})$ the set of plays whose payoff is strictly smaller than some payoff in P_{σ_0}.

[4] Recall that to have a solution to the SPS problem, memory is sometimes necessary as shown in Example 2.

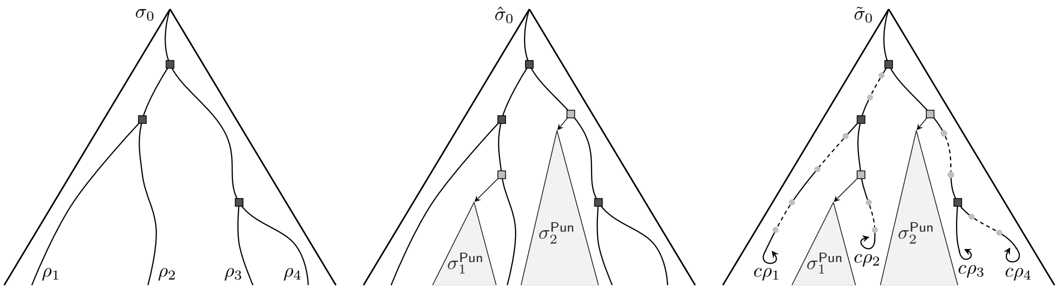

Figure 3 The creation of strategies $\hat{\sigma}_0$ and $\tilde{\sigma_0}$ from a solution σ_0 with $\mathsf{Wit}_{\sigma_0} = \{\rho_1, \rho_2, \rho_3, \rho_4\}$.

Deviations and Punishing Strategies. First, we define the set of deviations $\mathsf{Dev}(\mathsf{Wit}_{\sigma_0})$ as follows:

$$\mathsf{Dev}(\mathsf{Wit}_{\sigma_0}) = \{hv \in \mathsf{Hist}_{\sigma_0} \mid \mathsf{Wit}_{\sigma_0}(h) \neq \emptyset \land \mathsf{Wit}_{\sigma_0}(hv) = \emptyset\}.$$

As explained above, a deviation is a history that leaves the set of witnesses Wit_{σ_0} (by a move of Player 1).

Second, we establish the existence of canonical forms for punishing strategies. We potentially need an exponential number of them for reachability objectives and a polynomial number of them for parity objectives. In both cases, each punishing strategy has a size which can be bounded exponentially. The existence of those strategies are direct consequences of the two following lemmas.

▶ **Lemma 11** (Parity). *Let $v \in V$ be such that there exists $hv \in \mathsf{Dev}(\mathsf{Wit}_{\sigma_0})$. Then there exists a finite-memory strategy σ_v^{Pun} such that for all deviations $hv \in \mathsf{Dev}(\mathsf{Wit}_{\sigma_0})$, when Player 0 plays σ_v^{Pun} from hv, all consistent plays ρ starting in v are such that either $h\rho \in \Omega_0$ or $h\rho \in \Omega^{<}(P_{\sigma_0})$. The size of σ_v^{Pun} is at most exponential in the size of $\mathcal{G}$.*

Proof of Lemma 11. First, we note that, after a deviation $hv \in \mathsf{Dev}(\mathsf{Wit}_{\sigma_0})$, if Player 0 continues to play the strategy σ_0 from hv, then all consistent plays ρ are such that either $\rho \in \Omega_0$ or $\rho \in \Omega^{<}(P_{\sigma_0})$ as σ_0 is a solution to the SPS problem. Therefore, we know that Player 0 has a punishing strategy for all such deviations hv. Second, as parity objectives are prefix-independent, he can use one uniform strategy that only depends on v (and not on hv). There exists such a strategy with finite memory that can be constructed as follows. We express the objective $\Omega_0 \cup \Omega^{<}(P_{\sigma_0})$ as an explicit Muller objective [23] for a zero-sum game played on the arena G from initial vertex v. This objective is defined by the set $\{C \subseteq V \mid \exists \rho \text{ such that } \mathsf{Inf}(\rho) = C \land \rho \in \Omega_0 \cup \Omega^{<}(P_{\sigma_0})\}$. This exactly encodes the objective of Player 0 when he plays the punishing strategy after a deviation hv. It is well-known that in zero-sum explicit Muller games, there always exist finite-memory winning strategies with a size exponential in the number $|V|$ of vertices of the arena [16]. ◀

▶ **Lemma 12** (Reachability). *Let $v \in V$ and $(w, p) \in \{0,1\} \times \{0,1\}^t$ be such that there exists $hv \in \mathsf{Dev}(\mathsf{Wit}_{\sigma_0})$ with $(\mathsf{won}(hv), \mathsf{pay}(hv)) = (w, p)$. Then there exists a finite-memory strategy $\sigma_{(v,w,p)}^{\mathsf{Pun}}$ such that for all deviations $hv \in \mathsf{Dev}(\mathsf{Wit}_{\sigma_0})$ with $(\mathsf{won}(hv), \mathsf{pay}(hv)) = (w, p)$, when Player 0 plays $\sigma_{(v,w,p)}^{\mathsf{Pun}}$ from hv, all consistent plays ρ starting in v are such that either $h\rho \in \Omega_0$ or $h\rho \in \Omega^{<}(P_{\sigma_0})$. The size of $\sigma_{(v,w,p)}^{\mathsf{Pun}}$ is at most exponential in the size of $\mathcal{G}$.*

Proof of Lemma 12. We follow the same reasoning as in the proof of Lemma 11, except that reachability objectives are not prefix-independent. We thus need to take into account the set of objectives Ω_i already satisfied along the history hv, which is recorded in (w, p). The uniform finite-memory strategy $\sigma^{\mathsf{Pun}}_{(v,w,p)}$ that Player 0 can use from all deviations hv such that $\mathsf{won}(hv) = w$ and $\mathsf{pay}(hv) = p$ is constructed as follows. First, notice that if $w = 1$, meaning that objective Ω_0 is already satisfied, then Player 0 can play using any memoryless strategy as punishing strategy. Second, if $w = 0$, as done in Subsection 3.2, we consider the extension of G such that its vertices are of the form (v', w', p') where the (w', p')-component keeps track of the objectives that have been satisfied so far and such that its initial vertex is equal to (v, w, p). On this extended arena, we consider the zero-sum game with the objective $\Omega_0 \cup \Omega^{<}(P_{\sigma_0})$ encoded as the disjunction of a reachability objective (Ω_0) and a safety objective $(\Omega^{<}(P_{\sigma_0}))$. More precisely, in the extended game, Player 0 has the objective either to reach a vertex in the set $\{(v', w', p') \mid w' = 1\}$ or to stay forever within the set of vertices $\{(v', w', p') \mid \exists p'' \in P_{\sigma_0} : p' < p''\}$. It is known, see e.g. [9], that there always exist memoryless winning strategies for zero-sum games with an objective which is the disjunction of a reachability objective and a safety objective. Therefore, this is the case here for the extended game, and thus also in the original game however with a winning finite-memory strategy with exponential size. ◀

If we systematically change within σ_0 the behavior of Player 0 after a deviation from Wit_{σ_0}, and use the punishing strategies as defined in the proofs of Lemmas 11 and 12, we obtain a new strategy $\hat{\sigma}_0$ that is solution to the SPS problem. The total size of the punishing finite-memory strategies in $\hat{\sigma}_0$ is at most exponential in the size of $\mathcal{G}$. To obtain our results, it remains to show how to compact the plays in Wit_{σ_0}. To that end, we study the histories and plays within Wit_{σ_0}.

Compacting Witnesses. We now show how to compact the set of witnesses in a way to produce them with a finite-memory strategy. Together with the punishing strategies this will lead to a solution $\tilde{\sigma}_0$ to SPS problem with a memory of exponential size. We first consider reachability objectives and explain later how to modify the construction for parity objectives.

Given a history h that is prefix of at least one witness in Wit_{σ_0}, we call *region* and we denote by $\mathsf{Reg}(h)$ the tuple $\mathsf{Reg}(h) = (\mathsf{won}(h), \mathsf{pay}(h), \mathsf{Wit}_{\sigma_0}(h))$. We also use notation $R = (w, p, W)$ for a region. Given a witness $\rho = v_0 v_1 \ldots \in \mathsf{Wit}_{\sigma_0}$, we consider $\rho^* = (v_0, R_0)(v_1, R_1) \ldots$ such that each v_j is extended with the region $R_j = (w_j, p_j, W_j) = \mathsf{Reg}(v_0 v_1 \ldots v_j)$. Similarly we define h^* associated with any history h prefix of a witness. The following properties hold for a witness ρ and its corresponding play ρ^*:

- for all $j \geq 0$, we have $w_j \leq w_{j+1}$, $p_j \leq p_{j+1}$, and $W_j \supseteq W_{j+1}$,
- the sequence $(w_j, p_j)_{j \geq 0}$ eventually stabilizes on (w, p) equal to the extended payoff $(\mathsf{won}(\rho), \mathsf{pay}(\rho))$ of ρ,
- the sequence $(W_j)_{j \geq 0}$ eventually stabilizes on a set W which is a singleton such that $W = \{\rho\}$.

Thanks to the previous properties, each $\rho \in \mathsf{Wit}_{\sigma_0}$ can be *region decomposed* into a sequence of paths $\pi[1]\pi[2]\cdots\pi[k]$ where the corresponding decomposition $\pi^*[1]\pi^*[2]\cdots\pi^*[k]$ of ρ^* is such that for each ℓ: *(i)* the region is constant along the path $\pi^*[\ell]$ and *(ii)* it is distinct from the region of the next path $\pi^*[\ell+1]$ (if $\ell < k$). Each $\pi[\ell]$ is called a *section* of ρ, such that it is *internal* (resp. *terminal*) if $\ell < k$ (resp. $\ell = k$).

Notice that the number of regions that are traversed by ρ is bounded by

$$(t + 2) \cdot |\mathsf{Wit}_{\sigma_0}|. \tag{4}$$

Indeed along ρ, the first two components (w, p) of a region correspond to a monotonically increasing vector of $t + 1$ Boolean values (from $(0, (0, \ldots, 0))$ to $(1, (1, \ldots, 1))$ in the worst case), and the last component W is a monotonically decreasing set of witnesses (from Wit_{σ_0} to $\{\rho\}$ in the worst case). So the number of regions traversed by a witness is bounded exponentially in the size of the game $\mathcal{G}$.

We have the following important properties for the sections of the witnesses of Wit_{σ_0}.

- Let $\rho, \rho' \in \mathsf{Wit}_{\sigma_0}$, with region decompositions $\rho = \pi[1] \cdots \pi[k]$ and $\rho' = \pi'[1] \cdots \pi'[k']$ and let h be the longest common prefix of ρ and ρ'. Then there exists $k_1 < k, k'$ such that $h = \pi[1] \cdots \pi[k_1]$, $\pi[\ell] = \pi'[\ell]$ for all $\ell \in \{1, \ldots, k_1\}$ and $\pi[k_1 + 1] \neq \pi'[k_1 + 1]$. Therefore, when Wit_{σ_0} is seen as a tree, the branching structure of this tree is respected by the sections.
- Let $R = (w, p, W)$ be a region and consider the set of all histories h such that $\mathsf{Reg}(h) = R$. Then all these histories are prefixes of each other and are prefixes of exactly $|W|$ witnesses (as $\mathsf{Wit}_{\sigma_0}(h) = W$ for each such h). Therefore, the branching structure of Wit_{σ_0} is respected by the sections such that the associated regions are all pairwise distinct. The latter property is called the *region-tree structure* of Wit_{σ_0}.

We consider a *compact* version $c\mathsf{Wit}_{\sigma_0}$ of Wit_{σ_0} defined as follows:

- each internal section π of Wit_{σ_0} is replaced by the elementary path $c\pi$ obtained by eliminating all the cycles of π. Each terminal section π of Wit_{σ_0} is replaced by a lasso $c\pi = \pi'_1(u\pi'_2)^\omega$ such that u is a vertex, $\pi'_1 u \pi'_2$ is an elementary path, and $\pi'_1 u \pi'_2 u$ is prefix of π.
- each witness ρ of Wit_{σ_0} with region decomposition $\rho = \pi[1] \cdots \pi[k]$ is replaced by $c\rho = c\pi[1] \cdots c\pi[k]$ such that each $\pi[\ell]$ is replaced by $c\pi[\ell]$. Notice that as the region is constant inside the sections, the region decomposition of $c\rho$ coincide with the sequence of its $c\pi[\ell]$, $\ell \in \{1, \ldots, k\}$.

Therefore, by construction of the compact witnesses, the region-tree structure of Wit_{σ_0} is kept by the set $\{c\rho \mid \rho \in \mathsf{Wit}_{\sigma_0}\}$ and for each $c\rho \in c\mathsf{Wit}_{\sigma_0}$,

$$(\mathsf{won}(c\rho), \mathsf{pay}(c\rho)) = (\mathsf{won}(\rho), \mathsf{pay}(\rho)). \tag{5}$$

We then construct the announced strategy $\tilde{\sigma}_0$ that produces the set $c\mathsf{Wit}_{\sigma_0}$ of compact witnesses and after any deviation acts with the adequate punishing strategy (as mentioned in Lemma 12). More precisely, let gv be such that g is prefix of a compact witness and gv is not (Player 1 deviates from $c\mathsf{Wit}_{\sigma_0}$). Then by definition of the compact witnesses, there exists a deviation hv such that $(\mathsf{won}(gv), \mathsf{pay}(gv)) = (\mathsf{won}(hv), \mathsf{pay}(hv)) = (w, p)$. Then from gv Player 0 switches to the punishing strategy $\sigma^{\mathsf{Pun}}_{(v,w,p)}$.

▶ **Lemma 13.** *The strategy $\tilde{\sigma}_0$ is a solution to the SPS problem for reachability SP games and its size is bounded exponentially in the size of the game $\mathcal{G}$.*

Proof of Lemma 13. Let us first prove that $\tilde{\sigma}_0$ is a solution to the SPS problem. *(i)* By (5), the set of extended payoffs of plays in $c\mathsf{Wit}_{\sigma_0}$ is equal to the set of extended payoffs of witnesses in Wit_{σ_0}. This means that with $c\mathsf{Wit}_{\sigma_0}$, we keep the same set P_{σ_0} and the objective Ω_0 is satisfied along each compact witness. *(ii)* The punishing strategies used by $\tilde{\sigma}_0$ guarantee the satisfaction of the objective $\Omega_0 \cup \Omega^{<}(P_{\sigma_0})$ by Lemma 12. Therefore $\tilde{\sigma}_0$ is a solution to the SPS problem.

Let us now show that the memory size of $\tilde{\sigma}_0$ is bounded exponentially in the size of $\mathcal{G}$. *(i)* By Lemma 12, each punishing strategy used by $\tilde{\sigma}_0$ is of exponential size and the number of punishing strategies is exponential. *(ii)* To produce the compact witnesses, $\tilde{\sigma}_0$ keeps in memory the current region and produces in a memoryless way the corresponding compact section (which is an elementary path or lasso). Thus the required memory size for producing $c\mathsf{Wit}_{\sigma_0}$ is the number of regions. By (4), every play in $c\mathsf{Wit}_{\sigma_0}$ traverses at most an exponential number of regions and there is an exponential number of such plays (equal to $|P_{\sigma_0}|$). ◀

We now switch to parity SP games and state the following lemma whose proof follows the same line of arguments as those given for reachability objectives.

▶ **Lemma 14.** *The strategy $\tilde{\sigma}_0$ is a solution to the SPS problem for parity SP games and its size is bounded exponentially in the size of the game $\mathcal{G}$.*

Proof of Lemma 14. We highlight here the main differences from reachability SP games.

- As parity objectives are prefix-independent, we associate to each history h of a play $\rho \in \mathsf{Wit}_{\sigma_0}$ a singleton $\mathsf{Reg}(h) = \mathsf{Wit}_{\sigma_0}(h)$ instead of the triplet $(\mathsf{won}(h), \mathsf{pay}(h), \mathsf{Wit}_{\sigma_0}(h))$ as in the case of reachability. This is because $(\mathsf{won}(h), \mathsf{pay}(h))$ does not make sense for prefix-independent objectives.
- For the definition of the compact witnesses, we proceed identically as for reachability by simply removing cycles inside each section with the exception of terminal sections. Given the terminal section $\pi[k]$ of a witness $\rho \in \mathsf{Wit}_{\sigma_0}$, we replace it by a lasso $c\pi[k] = \pi_1'(\pi_2')^\omega$ such that $c\pi[k]$ and $\pi[k]$ start at the same vertex, $\mathsf{Occ}(c\pi[k]) = \mathsf{Occ}(\pi[k])$, $\mathsf{Inf}(c\pi[k]) = \mathsf{Inf}(\pi[k])$, and $|\pi_1'\pi_2'|$ is quadratic in $|V|$ [3, Proposition 3.1]. Therefore, by construction, the objectives Ω_i satisfied by a witness ρ are exactly the same as for its corresponding compact play $c\rho$.

We then construct the strategy $\tilde{\sigma}_0$ that produces the set $c\mathsf{Wit}_{\sigma_0}$ of compact witnesses and after any deviation gv from $c\mathsf{Wit}_{\sigma_0}$ acts with the adequate punishing strategy σ_v^{Pun} (as mentioned in Lemma 11). ◀

Lemmas 13 and 14 lead to Proposition 10. Using this proposition, we are now able to prove our result on the NEXPTIME-membership of reachability and parity SP games.

Proof of Theorem 9. We have established the existence of solutions to the SPS problem that use a finite memory bounded exponentially, both for reachability (Lemma 13) and for parity (Lemma 14) SP games. Let σ_0 be such a solution. As it is finite-memory, we can guess it as a Moore machine $\mathcal{M}$ with a set of memory states at most exponential in the size of $\mathcal{G}$.

Let us explain how to verify that the guessed solution σ_0 is a solution to the SPS problem for parity objectives, i.e., every play in $\mathsf{Plays}_{\sigma_0}$ which is σ_0-fixed Pareto-optimal satisfies the objective Ω_0 of Player 0. First, we construct the cartesian product $G \times \mathcal{M}$ of the arena G with the Moore machine $\mathcal{M}$ which is a graph whose infinite paths (starting from the initial vertex v_0 and the initial memory state) are exactly the plays consistent with σ_0. Second, to compute P_{σ_0}, we test for the existence of a play ρ in $G \times \mathcal{M}$ with a given payoff $p = \mathsf{pay}(\rho)$, beginning with the largest possible payoff $p = (1, \dots, 1)$ and finishing with the smallest possible one $p = (0, \dots, 0)$. Verifying this corresponds to deciding whether there exists a play that satisfies an intersection of parity objectives. The latter property can be checked in polynomial time in the size of $G \times \mathcal{M}$ [17]. Third, to check that each Pareto optimal play in $\mathsf{Plays}_{\sigma_0}$ satisfies Ω_0, we test for each $p \in P_{\sigma_0}$ whether there exists a play that satisfies the objectives Ω_i such that $p_i = 1$ as well as the objective $\mathsf{Plays}_G \setminus \Omega_0$. As the complement of a parity objective is again a parity objective, we use again the polynomial algorithm of [17]. As a consequence we have a NEXPTIME algorithm for parity SP games.

The case of reachability SP games is solved similarly with the following two differences. Concerning the second step, the existence of a play in $G \times \mathcal{M}$ that satisfies an intersection of reachability objectives can be checked in polynomial time by first extending this graph with a Boolean vector in $\{0, 1\}^t$ keeping track of the objectives of Player 1 already satisfied.

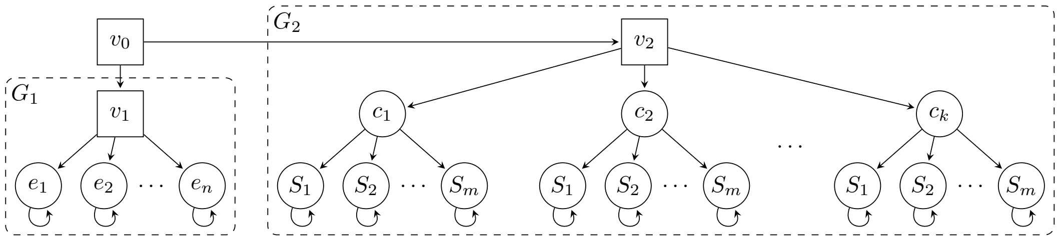

Figure 4 The tree arena used in the reduction from the SC problem.

Notice that the resulting graph is still of exponential size and that the intersection of reachability objectives becomes a single reachability objective. Concerning the third step, as the complement $\mathsf{Plays}_G \setminus \Omega_0$ of Ω_0 is not a reachability objective, we rather remove vertices of $G \times \mathcal{M}$ that contains an element of the target set T_0 before checking whether there exists a play that satisfies the objectives Ω_i such that $p_i = 1$ for a given $p \in P_{\sigma_0}$. ◀

4.2 NP-Completeness for Tree Arenas

We now turn to the NEXPTIME-hardness of the SPS problem. The proof of this result uses a reduction from the Succinct Set Cover problem. Before presenting this problem and proof in the next section, we want to show that the SPS problem is already NP-complete in the simple setting of reachability objectives and arenas that are trees. We use a reduction from the Set Cover problem (SC problem) which is NP-complete [24].

▶ **Theorem 15.** *The SPS problem is* NP*-complete for reachability SP games on tree arenas.*

Notice that when the game arena is a tree, it is easy to design an algorithm for solving the SPS problem that is in NP. First, we nondeterministically guess a strategy σ_0 that can be assumed to be memoryless as the arena is a tree. Second, we apply a depth-first search algorithm from the root vertex which accumulates to leaf vertices the extended payoff of plays which are consistent with σ_0. Finally, we check that σ_0 is a solution.

Let us explain why the SPS problem is NP-hard on tree arenas by reduction from the SC problem. We recall that an instance of the SC problem is defined by a set $C = \{e_1, e_2, \ldots, e_n\}$ of n elements, m subsets $S_1, S_2, \ldots, S_m$ such that $S_i \subseteq C$ for each $i \in \{1, \ldots, m\}$, and an integer $k \leq m$. The problem consists in finding k indexes $i_1, i_2, \ldots, i_k$ such that the union of the corresponding subsets equals C, i.e., $C = \bigcup\limits_{j=1}^{k} S_{i_j}$.

Given an instance of the SC problem, we construct a game with an arena consisting of $n + k \cdot (m + 1) + 3$ vertices. The arena G of the game is provided in Figure 4 and can be seen as two sub-arenas reachable from the initial vertex v_0. The game is such that there is a solution to the SC problem if and only if Player 0 has a strategy from v_0 in G which is a solution to the SPS problem. The game is played between Player 0 with reachability objective Ω_0 and Player 1 with $n+1$ reachability objectives. The objectives are defined as follows: $\Omega_0 = \mathsf{Reach}(\{v_2\})$, $\Omega_i = \mathsf{Reach}(\{e_i\} \cup \{S_j \mid e_i \in S_j\})$ for $i \in \{1, 2, \ldots, n\}$ and $\Omega_{n+1} = \mathsf{Reach}(\{v_2\})$. First, notice that every play in G_1 is consistent with any strategy of Player 0 and is lost by that player. It holds that for each $\ell \in \{1, 2, \ldots, n\}$, there is such a play with payoff $(p_1, \ldots, p_{n+1})$ such that $p_\ell = 1$ and $p_j = 0$ for $j \neq \ell$. These payoffs correspond to the elements e_ℓ we aim to cover in the SC problem. A play in G_2 visits v_2 and then a vertex c from which Player 0 selects a vertex S. Such a play is always won by Player 0 and its payoff is $(p_1, \ldots, p_{n+1})$ such that $p_{n+1} = 1$ and $p_r = 1$ if and only if the element e_r belongs to the set S. It follows that the payoff of such a play corresponds to a set of elements in the SC problem. It is easy to see that the following proposition holds and it follows that, as a consequence, Theorem 15 holds.

▶ **Proposition 16.** *There is a solution to an instance of the SC problem if and only if Player 0 has a strategy from v_0 in the corresponding SP game that is a solution to the SPS problem.*

Proof of Proposition 16. First, let us assume that there is a solution to the SC problem. It holds that there exists a set of k indexes $i_1, i_2, \dots, i_k$ such that the union of the corresponding sets equals the set C of elements we aim to cover. We define the strategy σ_0 as follows: $\sigma_0(v_0v_2c_j) = S_{i_j}$. Let us show that this strategy is solution to the SPS problem by showing that any play with a σ_0-fixed Pareto-optimal payoff is won by Player 0. This amounts to showing that for every play in G_1 there is a play in G_2 with a strictly larger payoff. This is sufficient as it makes sure that the payoff of plays in G_1 are not σ_0-fixed Pareto-optimal and as every play in G_2 is won by Player 0. Let $p = (p_1, \dots, p_{n+1})$ be the payoff of a play in G_1. It holds that $p_\ell = 1$ for some $\ell \in \{1, 2, \dots, n\}$ and $p_j = 0$ for $\ell \neq j$. This corresponds to the element e_ℓ in C. Since the k indexes $i_1, i_2, \dots, i_k$ are a solution to the SC problem, it holds that there exists some index i_j such that $e_\ell \in S_{i_j}$. It also holds that the play $v_0v_2c_j(S_{i_j})^\omega$ is consistent with σ_0. Its payoff is $p' = (p'_1, \dots, p'_{n+1})$ with $p'_\ell = 1$ since $e_\ell \in S_{i_j}$ and $p'_{n+1} = 1$. It follows that payoff p' is strictly larger than p.

Now, let us assume that Player 0 has a strategy σ_0 from v_0 that is a solution to the SPS problem. We can show that the set of indexes $\{i_j \mid \sigma_0(v_0v_2c_j) = S_{i_j}, j \in \{1, \dots, k\}\}$ is a solution to the SC problem. It is easy to see that since strategy σ_0 is a solution to the SPS problem, every payoff p in G_1 is strictly smaller than some payoff p' in G_2. It follows that in the SC problem, each element $e \in C$ corresponding to p is contained in some set S corresponding to p'. Since it also holds that $S \subseteq C$ for each set S, it follows that the sets mentioned above are an exact cover of C. ◀

4.3 NEXPTIME-Hardness

Let us come back to regular game arenas and show the NEXPTIME-hardness result for both reachability and parity SP games. Each type of objective is studied in a dedicated subsection.

▶ **Theorem 17.** *The SPS problem is* NEXPTIME*-hard for reachability SP games.*

▶ **Theorem 18.** *The SPS problem is* NEXPTIME*-hard for parity SP games.*

The NEXPTIME-hardness is obtained thanks to the succinct variant of the SC problem presented below.

4.3.1 Succinct Set Cover Problem

The *Succinct Set Cover problem (SSC problem)* is defined as follows. We are given a Conjunctive Normal Form (CNF) formula $\phi = C_1 \wedge C_2 \wedge \dots \wedge C_p$ over the variables $X = \{x_1, x_2, \dots, x_m\}$ made up of p clauses, each containing some disjunction of literals of the variables in X. The set of valuations of the variables X which satisfy ϕ is written $[\![\phi]\!]$. We are also given an integer $k \in \mathbb{N}$ (encoded in binary) and an other CNF formula $\psi = D_1 \wedge D_2 \wedge \dots \wedge D_q$ over the variables $X \cup Y$ with $Y = \{y_1, y_2, \dots, y_n\}$, made up of q clauses. Given a valuation $val_Y : Y \to \{0, 1\}$ of the variables in Y, called a *partial valuation*, we write $\psi[val_Y]$ the CNF formula obtained by replacing in ψ each variable $y \in Y$ by its valuation

$val_Y(y)$. We write $\llbracket\psi[val_Y]\rrbracket$ the valuations of the remaining variables X which satisfy $\psi[val_Y]$. The SSC problem is to decide whether there exists a set $K = \{val_Y \mid val_Y : Y \rightarrow \{0,1\}\}$ of k valuations of the variables in Y such that the valuations of the remaining variables X which satisfy the formulas $\psi[val_Y]$ include the valuations of X which satisfy ϕ. Formally, we write this $\llbracket\phi\rrbracket \subseteq \bigcup_{val_Y \in K} \llbracket\psi[val_Y]\rrbracket$.

We can show that this corresponds to a set cover problem succinctly defined using CNF formulas. The set $\llbracket\phi\rrbracket$ of valuations of X which satisfy ϕ corresponds to the set of elements we aim to cover. Parameter k is the number of sets that can be used to cover these elements. Such a set is described by a formula $\psi[val_Y]$, given a partial valuation val_Y, and its elements are the valuations of X in $\llbracket\psi[val_Y]\rrbracket$. This is illustrated in the following example.

▶ **Example 19.** Consider the CNF formula $\phi = (x_1 \vee \neg x_2) \wedge (x_2 \vee x_3)$ over the variables $X = \{x_1, x_2, x_3\}$. The set of valuations of the variables which satisfy ϕ is $\llbracket\phi\rrbracket = \{(1,1,1),(1,1,0),(1,0,1),(0,0,1)\}$. Each such valuation corresponds to one element we aim to cover. Consider the CNF formula $\psi = (y_1 \vee y_2) \wedge (x_1 \vee y_2) \wedge (x_2 \vee x_3 \vee y_1)$ over the variables $X \cup Y$ with $Y = \{y_1, y_2\}$. Given the partial valuation val_Y of the variables in Y such that $val_Y(y_1) = 0$ and $val_Y(y_2) = 1$, we get the CNF formula $\psi[val_Y] = (0 \vee 1) \wedge (x_1 \vee 1) \wedge (x_2 \vee x_3 \vee 0)$. This formula describes the contents of the set identified by the partial valuation (as a partial valuation yields a unique formula). The valuations of the variables X which satisfy $\psi[val_Y]$ are the elements contained in the set. In this case, these elements are $\llbracket\psi[val_Y]\rrbracket = \{(0,1,0),(0,0,1),(0,1,1),(1,1,0),(1,0,1),(1,1,1)\}$. We can see that this set contains the elements $\{(1,1,1),(1,1,0),(1,0,1),(0,0,1)\}$ of $\llbracket\phi\rrbracket$.

The following result is used in the proof of our NEXPTIME-hardness results and is of potential independent interest.

▶ **Theorem 20.** *The SSC problem is* NEXPTIME*-complete.*

Proof of Theorem 20. It is easy to see that the SSC problem is in NEXPTIME. We can show that the SSC problem is NEXPTIME-hard by reduction from the *Succinct Dominating Set problem* (SDS problem) which is known to be NEXPTIME-complete for graphs *succinctly* defined using CNF formulas [14]. An instance of the SDS problem is defined by a CNF formula θ over two sets of n variables $X = \{x_1, x_2, \ldots, x_n\}$ and $Y = \{y_1, y_2, \ldots, y_n\}$ and an integer k (encoded in binary). The formula θ succinctly defines an undirected graph in the following way. The set of vertices is the set of all valuations of the n variables in X (or over the n variables in Y) of which there are 2^n. Let val_X and val_Y be two such valuations, representing two vertices. Then, there is an edge between val_X and val_Y if and only if $\theta[val_X, val_Y]$ or $\theta[val_Y, val_X]$ is true. An instance of the SDS problem is positive if there exists a set $K = \{val^1_X, val^2_X, \ldots, val^k_X\}$ of k valuations of the variables in X, corresponding to k vertices, such that all vertices in the graph are adjacent to a vertex in K. Formally, we write this $|\bigcup_{val_X \in K} \{val_Y \mid \theta[val_X, val_Y] \vee \theta[val_Y, val_X] \text{ is true}\}| = 2^n$.

The SDS problem can be reduced in polynomial time to the SSC problem as follows. We define the CNF formula ϕ over the set of variables X such that the formula is empty. Therefore, the set $\llbracket\phi\rrbracket$ is equal to the 2^n valuations of the variables in X. We then define the CNF formula ψ over the set of variables X and Y such that it is the CNF equivalent to $\theta(X,Y) \vee \theta(Y,X)$. The latter formula has a size which is polynomial in the size of the CNF formula θ which defines the graph. We keep the same integer k. Then, it is direct to see that the instance of the SDS problem is positive if and only if the instance of SSC problem is positive. Indeed, there is a positive instance to the SDS problem if and only if there exists a set K of k valuations of the variables in Y such that $\llbracket\phi\rrbracket \subseteq \bigcup_{val_Y \in K} \llbracket\psi[val_Y]\rrbracket$. ◀

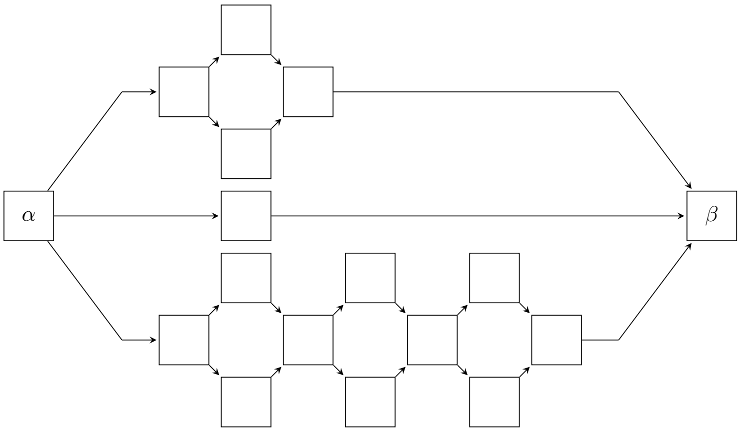

Figure 5 The gadget Q_{11}.

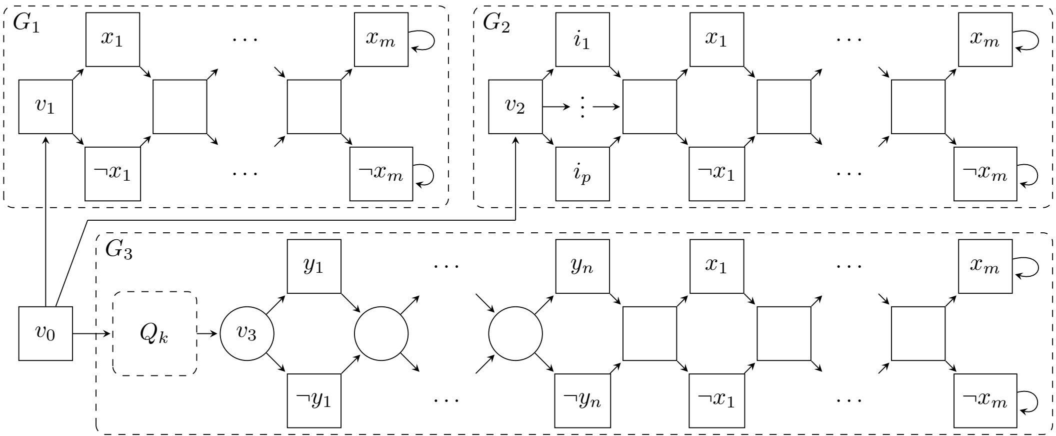

Figure 6 The arena G used in the reduction from the SSC problem.

4.3.2 NEXPTIME-Hardness of Reachability SP Games

We now describe in details our reduction from the SSC problem which allows us to show the NEXPTIME-hardness of solving the SPS problem in reachability SP games.

Gadget Q_k. Parameter k can be represented in binary using $r = \lfloor log_2(k) \rfloor + 1$ bits. It also holds that the binary encoding of k corresponds to the sum of at most r powers of 2. Given the binary encoding $b_0 b_1 \dots b_{r-1}$ of k such that $b_i \in \{0, 1\}$, let $ones = \{i \in \{0, \dots, r-1\} \mid b_i = 1\}$. It holds that $k = \sum_{i \in ones} 2^i$. Our gadget Q_k is a graph with a polynomial number of vertices (in the length of the binary encoding of k) such that all these vertices belong to Player 1. For each $i \in ones$ there is 2^i different paths from the initial vertex α to vertex β. Therefore, it holds that in Q_k there are k different paths from vertex α to vertex β.

▶ **Example 21.** Let $k = 11$, it holds that it can be represented in binary using $\lfloor log_2(11) \rfloor + 1 = 4$ bits. The binary representation of 11 is 1011 and it can be obtained by the following sum $2^3 + 2^1 + 2^0$. The gadget Q_{11} is detailed in Figure 5.

Game Arena. The arena G of the game is provided in Figure 6. It can be viewed as three sub-arenas that can be reached from the initial vertex v_0. We call these sub-arenas G_1, G_2

and G_3. Sub-arena G_3 starts with the gadget Q_k described previously. The number of vertices of this arena is polynomial in the number of clauses and variables in the formulas ϕ and ψ and in the length of the binary encoding of the integer k. The game is played between Player 0 with reachability objective Ω_0 and Player 1 with $t = 1 + 2m + p + q$ reachability objectives. The payoff of a play therefore consists in a single Boolean for objective Ω_1, a vector of $2m$ Booleans for objectives $\Omega_{x_1}, \Omega_{\neg x_1}, \ldots, \Omega_{x_m}, \Omega_{\neg x_m}$, a vector of p Booleans for objectives $\Omega_{C_1}, \ldots, \Omega_{C_p}$ and a vector of q Booleans for objectives $\Omega_{D_1}, \ldots, \Omega_{D_q}$.

Objectives. Each objective Ω_i in the game is a reachability objective $\mathsf{Reach}(T_i)$ defined by a target set T_i as follows. We later explain how these objectives are used in our reduction.

- The objective Ω_0 of Player 0 and objective Ω_1 of Player 1 are $\mathsf{Reach}(\{v_2, \alpha\})$ where α is the initial vertex of gadget Q_k.
- The target set for the objective Ω_{x_i} (resp. $\Omega_{\neg x_i}$) is the set of vertices $\{x_i\}$ (resp. $\{\neg x_i\}$) in G_1, G_2 and G_3.
- The target sets for the objective Ω_{C_i} with $i \in \{1, \ldots, p\}$ is the set of vertices in G_1 and G_3 corresponding to the literals of X which make up the clause C_i in ϕ. In addition, vertex i_j in G_2 belongs to the target set of objective Ω_{C_ℓ} for all $\ell \in \{1, \ldots, p\}$ such that $\ell \neq j$.
- The target set of objective Ω_{D_i} with $i \in \{1, \ldots, q\}$ is the set of vertices in G_3 corresponding to the literals of X and Y which make up the clause D_i in ψ. In addition, vertices v_1 and v_2 satisfy every objective Ω_{D_i} with $i \in \{1, \ldots, q\}$.

Payoff of Plays in G_1 and G_2. In each sub-arena G_1 and G_2, for each variable $x_i \in X$, there is one choice vertex controlled by Player 1 which leads to x_i and $\neg x_i$. These vertices have the next choice vertex as their successor, except for vertices x_m and $\neg x_m$ which have a self loop. In G_2, there is also a vertex v_2 controlled by Player 1 with p successors, each leading to the first choice vertex for the variables in X. The sub-arenas G_1 and G_2 are completely controlled by Player 1. The plays in the corresponding sub-arenas are therefore consistent with any strategy of Player 0.

The plays in G_1 do not satisfy objective Ω_0 of Player 0 nor objective Ω_1 of Player 1. A play in G_1 is of the form $v_1 \,\square\, z_1 \,\square\, z_2 \cdots \square\, (z_m)^\omega$ where z_i is either x_i or $\neg x_i$. It follows that a play satisfies the objective Ω_{x_i} or $\Omega_{\neg x_i}$ for each $x_i \in X$. The vector of payoffs for these objectives corresponds to a valuation of the variables in X. In addition, due to the way the objectives are defined, objective Ω_{C_i} is satisfied in a play if and only if clause C_i of ϕ is satisfied by the valuation this play corresponds to. The objective Ω_{D_i} for $i \in \{1, \ldots, q\}$ is satisfied in every play in G_1.

▶ **Lemma 22.** *The plays in G_1 are consistent with any strategy of Player 0. The payoffs of plays in G_1 are of the form $(0, val, sat(\phi, val), 1, \ldots, 1)$ where val is a valuation of the variables in X expressed as a vector of payoffs for the objectives Ω_{x_1} to $\Omega_{\neg x_m}$ and $sat(\phi, val)$ is the vector of payoffs for objectives Ω_{C_1} to Ω_{C_p} corresponding to that valuation. It holds that no plays in G_1 are won by Player 0.*

The plays in G_2 satisfy the objectives Ω_0 of Player 0 and Ω_1 of Player 1. A play in G_2 is of the form $v_2 i_j \,\square\, z_1 \,\square\, z_2 \cdots \square\, (z_m)^\omega$ where z_ℓ is either x_ℓ or $\neg x_\ell$. It follows that a play satisfies either the objective Ω_x or $\Omega_{\neg x}$ for each $x \in X$ which again corresponds to a valuation of the variables in X. The objective Ω_{D_i} for $i \in \{1, \ldots, q\}$ is satisfied in every play in G_2. Compared to the plays in G_1, the difference lies in the objectives corresponding to clauses of ϕ which are satisfied. In any play in G_2, a vertex i_j with $j \in \{1, \ldots, p\}$ is first visited, satisfying all the objectives Ω_{C_ℓ} with $\ell \in \{1, \dots, p\}$ and $\ell \neq j$. All but one objective corresponding to the clauses of ϕ are therefore satisfied.

▶ **Lemma 23.** *The plays in G_2 are consistent with any strategy of Player 0. The payoffs of plays in G_2 are of the form $(1, val, vec, 1, \dots, 1)$ where val is a valuation of the variables in X expressed as a vector of payoffs for objectives Ω_{x_1} to $\Omega_{\neg x_m}$ and vec is a vector of payoffs for objectives Ω_{C_1} to Ω_{C_p} in which all of them except one are satisfied. It holds that all plays in G_2 are won by Player 0.*

From the two previous lemmas, we can state the following lemma when considering the payoffs of plays in G_1 and G_2.

▶ **Lemma 24.** *For every play in G_1 which corresponds to a valuation of the variables in X that does not satisfy ϕ, there is a play in G_2 with a strictly larger payoff.*

Proof of Lemma 24. Let ρ be a play in G_1 which corresponds to a valuation of the variables in X that does not satisfy ϕ. It follows that at least one objective, say Ω_{C_ℓ}, is not satisfied in ρ as at least one clause of ϕ (clause C_ℓ) is not satisfied by that valuation. Let us consider the play ρ' in G_2 which visits vertex i_ℓ and after visits the vertices corresponding to the same valuation of the variables in X as ρ. By Lemmas 22 and 23, it follows that the payoff of ρ' is strictly larger than that of ρ (as we have $(0, val, sat(\phi, val), 1, \dots, 1) < (1, val, vec, 1, \dots, 1)$ with $sat(\phi, val) \leq vec$). ◀

The following lemma is a consequence of Lemma 24.

▶ **Lemma 25.** *The set of payoffs of plays in G_1 that are σ_0-fixed Pareto-optimal when considering $G_1 \cup G_2$ for any strategy σ_0 of Player 0 is equal to the set of payoffs of plays in G_1 whose valuation of X satisfy ϕ.*

Proof of Lemma 25. This property stems from the following observations. First, any play in G_1 which satisfies every objective Ω_{C_i} with $i \in \{1, \dots, p\}$, and therefore corresponds to a valuation of X which satisfies ϕ, has a payoff that is incomparable to every possible payoff in G_2. This is because such a play satisfies more objectives in $\Omega_{C_1}, \dots, \Omega_{C_p}$ than the plays in G_2 but does not satisfy objective Ω_1 while the plays in G_2 do. Second, every other play in G_1 has a strictly smaller payoff then at least one play in G_2 due to Lemma 24 and its payoff is therefore not σ_0-fixed Pareto-optimal. ◀

Problematic Payoffs in G_1. The plays described in the previous lemma correspond exactly to the valuations of X which satisfy ϕ and therefore to the elements we aim to cover in the SSC problem. They are σ_0-fixed Pareto-optimal when considering $G_1 \cup G_2$ and are lost by Player 0. All other σ_0-fixed Pareto-optimal payoffs in $G_1 \cup G_2$ are only realized by plays in G_2 which are all won by Player 0. It follows that in order for Player 0 to find a strategy σ_0 from v_0 that is solution to the SPS problem, it must hold that those payoffs are not σ_0-fixed Pareto-optimal when considering $G_1 \cup G_2 \cup G_3$. Otherwise, a play consistent with σ_0 with a σ_0-fixed Pareto-optimal payoff is lost by Player 0. We call those payoffs *problematic payoffs.*

Creating Strictly Larger Payoffs in G_3. In order for Player 0 to find a strategy σ_0 which is a solution to the SPS problem, this strategy must be such that for each problematic payoff in G_1, there is a play in G_3 consistent with σ_0 and with a strictly larger payoff. Since the plays in G_3 are all won by Player 0, this would ensure that the strategy σ_0 is a solution to the problem. This corresponds in the SSC problem to selecting a series of sets in order to cover the valuations of X which satisfy ϕ.

Payoff of Plays in G_3. Sub-arena G_3 starts with gadget Q_k whose vertices are controlled by Player 1. Then, for each variable $y_i \in Y$, there is one choice vertex controlled by Player 0 which leads to y_i and $\neg y_i$. These vertices have the next choice vertex as their successor, except for y_n and $\neg y_n$ which lead to the first choice vertex for the variables in X. Each play in G_3 satisfies the objectives Ω_0 of Player 0 and Ω_1 of Player 1. A play in G_3 consistent with a strategy σ_0 is of the form $(\alpha \square \cdots \square \beta)(\bigcirc r_1 \bigcirc r_2 \dots \bigcirc r_n)(\square z_1 \square z_2 \cdots \square (z_m)^\omega)$ where r_i is either y_i or $\neg y_i$ and z_i is either x_i or $\neg x_i$. Since only the choice vertices leading to y or $\neg y$ for $y \in Y$ belong to Player 0, it holds that $(\bigcirc r_1 \bigcirc r_2 \dots \bigcirc r_n)$ is the only part of any play in G_3 which is directly influenced by σ_0. Since that part of a play comes after a history from α to β of which there are k and by definition of a strategy, this can be interpreted as choosing k valuations of the variables in Y. After this, the play satisfies either the objective Ω_x or $\Omega_{\neg x}$ for each $x \in X$ which corresponds to a valuation of X. Due to the way the objectives are defined, the objective Ω_{C_i} (resp. Ω_{D_i}) is satisfied if and only if clause C_i of ϕ (resp. D_i of ψ) is satisfied by the valuation the play corresponds to. In order to create a play with a payoff r' that is strictly larger than a problematic payoff r, σ_0 must choose a valuation of Y such that there exists a valuation of the remaining variables X which together with this valuation of Y satisfies ψ and ϕ (since in r every objective Ω_{C_i} for $i \in \{1, \dots, p\}$ and Ω_{D_i} for $i \in \{1, \dots, q\}$ is satisfied). Since the plays in G_3 also satisfy the objective Ω_1 and plays in G_1 do not, this ensures that $r < r'$.

We can finally establish that our reduction is correct.

▶ **Proposition 26.** *Player 0 has a strategy σ_0 from v_0 in G that is a solution to the SPS problem if and only if there is a solution to the corresponding instance of the SSC problem.*

Proof of Proposition 26. Let us assume that that σ_0 is a solution to the SPS problem in G and show that there is a solution to the SSC problem. Let val_X be a valuation of the variables in X which satisfies ϕ. This valuation corresponds to a play in G_1 with a problematic payoff r. Since the objective of Player 0 is not satisfied in that play and since σ_0 is a solution to the SPS problem, it holds that r is not σ_0-fixed Pareto-optimal. It follows that there exists a play in G_3 that is consistent with σ_0 and whose payoff is strictly larger than r. As described above, such a play corresponds to a valuation val_Y of the variables in Y such that $val_X \in [\![\psi[val_Y]]\!]$. Since this can be done for each $val_X \in [\![\phi]\!]$ and since there is a set K of k possible valuations val_Y in G_3, it holds that $[\![\phi]\!] \subseteq \bigcup\limits_{val_Y \in K} [\![\psi[val_Y]]\!]$.

Let us now assume that there is a solution to the SSC problem and show that we can construct a strategy σ_0 that is solution to the SPS problem. Let K be the set of k valuations val_Y of the variables in Y which is a solution to the SSC problem. Since there are k possible histories from α to β in G_3, we define σ_0 such that the n vertices y_i or $\neg y_i$ for $i \in \{1, \dots, n\}$ visited after each history correspond to a valuation in K. We can now show that this strategy is a solution to the SPS problem. We do this by showing that each play ρ with problematic payoff r in G_1 has a strictly smaller payoff than that of some play ρ' with payoff r' in G_3. Such a payoff r corresponds to a valuation $val_X \in [\![\phi]\!]$. Since K is a solution to the SSC problem, it holds that there exists some valuation $val_Y \in K$ such that $val_X \in [\![\psi[val_Y]]\!]$. It follows, given the definition of σ_0, that there exists a play ρ' in G_3 corresponding to that valuation val_Y and which visits the vertices x or $\neg x$ for each $x \in X$ such that it corresponds to the valuation val_X. Given the properties mentioned before, the payoff r' of this play is such that $r < r'$. ◀

The previous proof yields our result on the NEXPTIME-hardness of the SPS problem in reachability SP games.

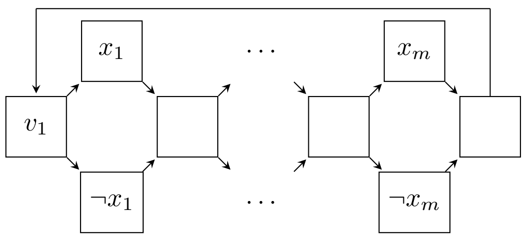

Figure 7 The repeating structure used in the reduction from the SSC problem for parity SP games.

4.3.3 NEXPTIME-Hardness of Parity SP Games

We now provide the NEXPTIME-hardness result for parity SP games.

The proof of this result follows the same ideas used in the proof for reachability SP games. It again uses a reduction from the SSC problem in which we construct an arena G, and its structure of three sub-arenas G_1, G_2, and G_3 is kept. We describe the main difficulties that we encounter when adapting this proof for parity objectives and how to overcome them by modifying each sub-arena G_i into G'_i. The modified arena G' is depicted in Figure 8.

Two Particular Objectives. Remember that for the case of reachability SP games, the objective Ω_0 of Player 0 and the first objective Ω_1 of Player 1 were either always satisfied or always not satisfied in all plays of a given sub-arena G_1, G_2, or G_3. This property holds in the modified arena G', with these objectives being expressed using parity conditions.

On the Choice of Valuations. Recall that for reachability objectives, we encoded the choice of valuations for the variables $x_i \in X$ made by Player 1 by using sub-arena G_1 of Figure 6 (which was also reused as part of G_2 and G_3). A play in this sub-arena encodes a valuation by visiting one literal $l_i \in \{x_i, \neg x_i\}$ for each $x_i \in X$ thus satisfying the objectives Ω_{l_i}, $i \in \{1, \ldots, m\}$.

This simple schema cannot be reused in the case of parity objectives as they are prefix-independent objectives. Instead, we ask Player 1 to repeatedly produce the *same choice* along loops (from v_1 back to v_1) in the adapted gadget of Figure 7.

We associate with each variable $x_i \in X$ two parity objectives with priority function c_{l_i} with $l_i \in \{x_i, \neg x_i\}$ defined as follows: $c_{l_i}(l_i) = 2$, $c_{l_i}(\neg l_i) = 1$ and $c_{l_i}(v) = 3$ for all the other vertices. It is easy to see that plays in which the valuation changes infinitely many times (for example, visiting x_i and $\neg x_i$ infinitely often), have a payoff strictly smaller than some other play which settles on a choice of valuation for each variable. Indeed, the payoff for objectives $\Omega_{x_i^1}$ and $\Omega_{x_i^2}$ is $(0, 0)$ in the first case and $(1, 0)$ or $(0, 1)$ in the second. We say that a valuation is *properly encoded* if Player 1 eventually repeats the same choice along the loops to settle on a valuation.

Notice that the gadget of Figure 7 appears nearly identical as part of the three sub-arenas G'_1, G'_2, and G'_3 of Figure 8. In all of these sub-arenas, we define the values of each priority function c_{l_i} with $l_i \in \{x_i, \neg x_i\}$ exactly as explained above.

On the Satisfied Clauses. Recall that in the case of reachability SP games, we associated one reachability objective with each clause C_i (resp. D_i) of ϕ (resp. ψ). As encoding valuations is made more complex by the prefix-independency of parity objectives, we also need to adapt the way we check which clauses are satisfied by a given valuation.

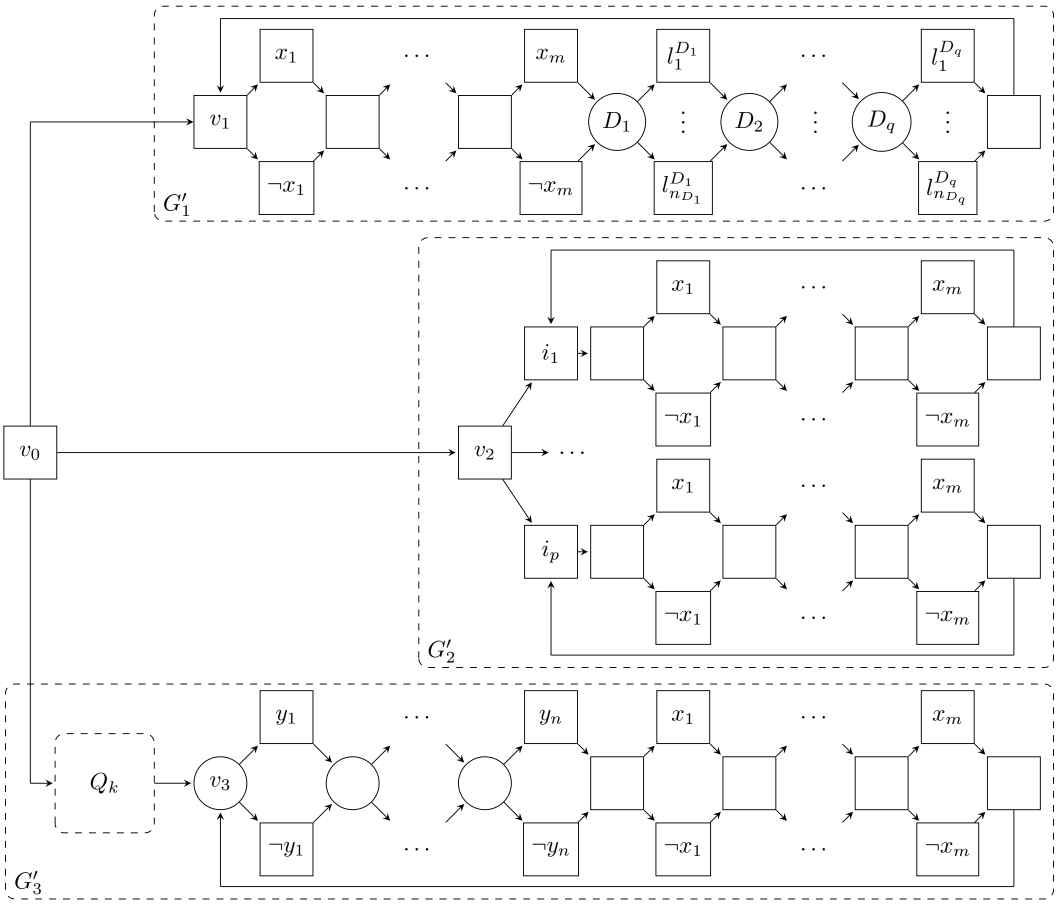

Figure 8 The arena G' used in the reduction from the SSC problem for parity SP games.

Let us first explain our encoding for clauses C_j of ϕ. We associate one parity objective with priority function $c_l^{C_j}$ with *each literal* l of each clause C_j. Therefore, if $C_j = l_1^{C_j} \vee \ldots \vee l_{n_{C_j}}^{C_j}$, there are n_{C_j} parity objectives for clause C_j. Priority function $c_l^{C_j}$ is defined as follows for each vertex $l_i \in \{x_i, \neg x_i\}$ that appears in G'_1 and G'_3 (we will define it later for G'_2): $c_l^{C_j}(l_i) = 2$ if $l = l_i$ and $c_l^{C_j}(l_i) = 1$ if $l = \neg l_i$, and for all the other vertices v, we have $c_l^{C_j}(v) = 3$. This encoding of clauses has the following important property: given a valuation val_X of the variables in X properly encoded by Player 1, a clause C_j is satisfied by val_X if and only if the parity condition $\mathsf{Parity}(c_l^{C_j})$ is satisfied for *at least one* of the literals l of C_j. Thus with the proposed encoding with priority functions, there are several ways to observe that a clause C_j is satisfied (the corresponding payoff is a *non-null* vector of n_{C_j} Booleans).

In the sub-arena G'_3, we see a part resembling the gadget of Figure 7, however made of vertices y_i, $i \in \{1, \ldots, n\}$. This part is related to the choice of a valuation of the variables in Y made by Player 0 with respect to ψ. To encode clauses D_j of ψ, we proceed exactly as we did previously with clauses C_j of ϕ. We associate one priority function $c_l^{D_j}$ with each literal l of each clause D_j (recall that such a literal uses both sets of variables X and Y). We similarly define the values of $c_l^{D_j}$ for vertices of G'_3: for $l' \in \{x_i, \neg x_i \mid i \in \{1, \ldots, m\}\} \cup \{y_i, \neg y_i \mid i \in \{1, \ldots, n\}\}$, we define $c_l^{D_j}(l') = 2$ if $l = l'$ and $c_l^{D_j}(l') = 1$ if $l = \neg l'$, and for all the other vertices v of G'_3, we define $c_l^{D_j}(v) = 3$. Notice that the definition of $c_l^{D_j}$ is given for G'_3 only. We will later give its definition for G'_1 and G'_2.

Modifications Needed on G_1. Let us now explain how to modify G_1 into G'_1. Remember that in the case of reachability SP games, the objective associated with each clause D_j of ψ is satisfied by all plays in G_1. Indeed the purpose of G_1 (in combination with G_2) is to isolate all the encodings of the valuations of X that satisfy ϕ independently of ψ. In case of a positive instance of the SSC problem, the payoff of these encodings are strictly smaller than that of some play in G_3, which have to satisfy all clauses of ψ by definition of this problem.

We proceed similarly in G'_1. However as there are several ways to satisfy D_j in ψ (at least one of its literals has to be satisfied), we let Player 0 choose which way to do it. This is encoded by the part of G'_1 made with vertices D_j, $j \in \{1, \ldots, q\}$, controlled by Player 0, and their successors $l_1^{D_j}, \ldots, l_{n_{D_j}}^{D_j}$ such that $D_j = l_1^{D_j} \vee \ldots \vee l_{n_{D_j}}^{D_j}$. Given a properly encoded X valuation val_X made by Player 1, Player 0 chooses a Y valuation val_Y such that if $val_X \in [\![\phi]\!]$, then $val_X \in [\![\psi[val_Y]]\!]$. Player 0 makes such a choice by selecting at least one literal l^{D_j} of D_j, for each clause D_j of ψ, such that l is satisfied by the valuation made of val_X and val_Y. This is encoded in G'_1 by defining the priority function $c_l^{D_j}$ such that $c_l^{D_j}(l) = 2$ and $c_l^{D_j}(v) = 3$ for all the other vertices v of G'_1.

Modifications Needed on G_2. Recall that in the case of reachability SP games, the construction of G_2 ensures that the plays of G_1 whose payoff is not strictly smaller than the payoff of a play in G_2 are exactly the plays of G_1 that encode X valuations val_X such that $val_X \in [\![\phi]\!]$. As a consequence, the objectives that are satisfied by the plays in G_2 are exactly *(i)* those associated with a valuation val_X, *(ii)* the objectives associated with all clauses D_j of ψ, and *(iii)* the objectives associated with all clauses C_j of ϕ except one.

We achieve the same requirement for parity SP games by using G'_2 in place of G_2. In this sub-arena, Player 1 first selects one clause C in ϕ and then in the selecting part of G'_2, the priority functions are defined as follows.

- We use the priority functions c_{l_i} with $l_i \in \{x_i, \neg x_i \mid i \in \{1, \ldots, m\}\}$ as defined above for encoding X valuations.

- The priority functions $c_l^{D_j}$ are all defined such that the associated objective $\mathsf{Parity}(c_l^{D_j})$ is satisfied.
- Similarly the priority functions $c_l^{C_j}$ are defined such that the associated objective $\mathsf{Parity}(c_l^{C_j})$ is satisfied, except for all priority functions $c_l^{C_j}$ such that $C_j = C$ for which this objective is not satisfied.

Modifications Needed on G_3. We have modified G_1 into G_1' and G_2 into G_2' such that the only plays in G_1' with a Pareto-optimal payoff when considering $G_1' \cup G_2'$, given any strategy of Player 0 are those that encode valuations $val_X \in [\![\phi]\!]$. In G_1', after such a valuation chosen by Player 1, Player 0 indicates which valuation val_Y to use such that $val_X \in [\![\psi[val_Y]]\!]$. He chooses this valuation val_Y by indicating for each clause D_j which literals of D_j he has chosen such that the valuation made of val_X and val_Y satisfies D_j.

Let us now explain how to modify G_3 into G_3'. After each of the k histories produced by gadget Q_k, both players have to choose some valuation (resp. val_X and val_Y) for the variables that they control. In case of a positive instance of SSC problem, Player 0 will be able to select one of the valuations val_Y that he used in G_1' such that $val_X \in [\![\psi[val_Y]]\!]$ whenever $val_X \in [\![\phi]\!]$. It follows that plays in G_3' have a larger payoff than the Pareto-optimal payoffs of $G_1' \cup G_2'$. In case of a negative instance of SSC problem, Player 0 will not be able to do so.

Clearly there exists a solution to the SPS problem in the modified arena G' if and only if the instance of the SSC problem is positive.

5 Conclusion

We have introduced in this paper the class of two-player SP games and the SPS problem in those games. We provided a reduction from SP games to a two-player zero-sum game called the C-P game in order to provide FPT results on solving this problem. We then showed how the arena and the generic objective of this C-P game can be adapted to specifically handle reachability and parity SP games. This allowed us to prove that reachability (resp. parity) SP games are in FPT when the number t of objectives of Player 1 (resp. when t and the maximal priority according to each priority function in the game) is a parameter. We then turned to the complexity class of the SPS problem and provided a proof of its NEXPTIME-membership, which relied on showing that any solution to the SPS problem in a reachability or parity SP game can be transformed into a solution with an exponential memory. We provided a proof of the NP-completeness of the problem in the simple setting of reachability SP games played on tree arenas. We then came back to regular game arenas and provided the proof of the NEXPTIME-hardness of the SPS problem in reachability and parity SP games. This proof relied on a reduction from the SSC problem which we proved to be NEXPTIME-complete, a result of potential independent interest.

In future work, we want to study other ω-regular objectives as well as quantitative objectives such as mean-payoff in the framework of SP games and the SPS problem. It would also be interesting to study whether other works, such as rational synthesis, could benefit from the approaches used in this paper.

References

1 Mrudula Balachander, Shibashis Guha, and Jean-François Raskin. Stackelberg mean-payoff games with a rationally bounded adversarial follower. *CoRR*, abs/2007.07209, 2020. URL: `https://arxiv.org/abs/2007.07209`, `arXiv:2007.07209`.

2 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.

3 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.

4 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.

5 Romain Brenguier, Jean-François Raskin, and Ocan Sankur. Assume-admissible synthesis. In Luca Aceto and David de Frutos-Escrig, editors, *26th International Conference on Concurrency Theory, CONCUR 2015, Madrid, Spain, September 1.4, 2015*, volume 42 of *LIPIcs*, pages 100–113. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2015. doi:10.4230/LIPIcs.CONCUR.2015.100.

6 Léonard Brice, Jean-François Raskin, and Marie van den Bogaard. Subgame-perfect equilibria in mean-payoff games. *CoRR*, abs/2101.10685, 2021. URL: https://arxiv.org/abs/2101.10685, arXiv:2101.10685.

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

8 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.

9 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.

10 Krishnendu Chatterjee and Monika Henzinger. Efficient and dynamic algorithms for alternating Büchi games and maximal end-component decomposition. *J. ACM*, 61(3):15:1–15:40, 2014. doi:10.1145/2597631.

11 Krishnendu Chatterjee and Thomas A. Henzinger. Assume-guarantee synthesis. In Orna Grumberg and Michael Huth, editors, *Tools and Algorithms for the Construction and Analysis of Systems, 13th International Conference, TACAS 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 4424 of *Lecture Notes in Computer Science*, pages 261–275. Springer, 2007. doi:10.1007/978-3-540-71209-1_21.

12 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.

13 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.

14 Bireswar Das, Patrick Scharpfenecker, and Jacobo Torán. CNF and DNF succinct graph encodings. *Inf. Comput.*, 253:436–447, 2017. doi:10.1016/j.ic.2016.06.009.

15 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.

16 Stefan Dziembowski, Marcin Jurdzinski, and Igor Walukiewicz. How much memory is needed to win infinite games? In *Proceedings, 12th Annual IEEE Symposium on Logic in Computer Science, Warsaw, Poland, June 29 - July 2, 1997*, pages 99–110. IEEE Computer Society, 1997. doi:10.1109/LICS.1997.614939.

17 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.

18 Emmanuel Filiot, Raffaella Gentilini, and Jean-François Raskin. The adversarial Stackelberg value in quantitative games. In Artur Czumaj, Anuj Dawar, and Emanuela Merelli, editors, *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020, July 8-11, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 168 of *LIPIcs*, pages 127:1–127:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.ICALP.2020.127.

19 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.

20 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.

21 Anshul Gupta and Sven Schewe. Quantitative verification in rational environments. In Amedeo Cesta, Carlo Combi, and François Laroussinie, editors, *21st International Symposium on Temporal Representation and Reasoning, TIME 2014, Verona, Italy, September 8-10, 2014*, pages 123–131. IEEE Computer Society, 2014. doi:10.1109/TIME.2014.9.

22 Anshul Gupta, Sven Schewe, and Dominik Wojtczak. Making the best of limited memory in multi-player discounted sum games. In Javier Esparza and Enrico Tronci, editors, *Proceedings Sixth International Symposium on Games, Automata, Logics and Formal Verification, GandALF 2015, Genoa, Italy, 21-22nd September 2015*, volume 193 of *EPTCS*, pages 16–30, 2015. doi:10.4204/EPTCS.193.2.

23 Florian Horn. Explicit Muller games are PTIME. In Ramesh Hariharan, Madhavan Mukund, and V. Vinay, editors, *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2008, December 9-11, 2008, Bangalore, India*, volume 2 of *LIPIcs*, pages 235–243. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2008. doi:10.4230/LIPIcs.FSTTCS.2008.1756.

24 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.

25 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.

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

27 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.

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

29 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.

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

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

32 Heinrich Freiherr von Stackelberg. *Marktform und Gleichgewicht.* Wien und Berlin, J. Springer, Cambridge, MA, 1937.

A Useful Result on SP Games

▶ **Proposition 27.** *Every parity (resp. reachability) SP game $\mathcal{G}$ with arena G containing n vertices can be transformed into a parity (resp. reachability) SP game $\bar{\mathcal{G}}$ with arena $\bar{G}$ containing at most n^2 vertices such that any vertex in $\bar{G}$ has at most 2 successors and Player 0 has a strategy σ_0 that is solution to the SPS problem in G if and only if Player 0 has a strategy $\bar{\sigma}_0$ that is solution to the problem in $\bar{G}$.*

Proof of Proposition 27. Let $\mathcal{G}$ be an SP game with arena G. Let us first describe the arena $\bar{G}$ of $\bar{\mathcal{G}}$. Let $v \in V$ be a vertex of G, then v is also a vertex of $\bar{G}$ such that it belongs to the same player and is the root of a complete binary tree with $\ell = |\{v' \mid (v, v') \in E\}|$ leaves if $(v, v) \notin E$. Otherwise, v has a self loop and its other successor is the root of such a tree with $\ell - 1$ leaves. The internal vertices of the tree (that is vertices which are not v, nor the leaves) belong to the same player as v. Each leaf vertex v' of this tree is such that $(v, v') \in E$, belongs to the same player as in G and is again the root of its own tree. The initial vertex v_0 of G remains unchanged in $\bar{G}$. Since every vertex in $\bar{G}$ is part of a binary tree or has a self loop and a single successor, it holds that it has at most two successors. Since G is a game arena, this transformation is such that each vertex in $\bar{G}$ has at least one successor. It follows that $\bar{G}$ is a game arena containing n vertices $v \in V$ and at most $n - 1$ internal vertices per tree in the case where $v \in V$ has n successors in G. It follows that the number of vertices in $\bar{G}$ is at most $n + n \cdot (n - 1) = n^2$. If $\mathcal{G}$ is a reachability SP game, the target sets remain unchanged in $\bar{G}$. For parity SP games, the priority function c remains unchanged for vertices $v \in V$ and we define $c(v') = c(v)$ for $v' \in \bar{V} \setminus V$ such that v' is an internal vertex of a tree whose root is v.

Let us now show that there is a solution to the SPS problem in $\mathcal{G}$ if and only if there is a solution in $\bar{\mathcal{G}}$. From each root v of a tree in $\bar{G}$ (corresponding to a vertex v of Player i in G) there is a set of $\ell = |\{v' \mid (v, v') \in E\}|$ different paths controlled by Player i, each leading to a vertex v'. It follows that there exists a play $\rho = v_0 v_1 v_2 \ldots \in \mathsf{Plays}_G$ if and only if there exists a play $\rho' = v_0 a_0 \ldots a_{n_1} v_1 b_0 \ldots b_{n_2} v_2 \ldots \in \mathsf{Plays}_{\bar{G}}$ such that every vertex a_i (resp. b_i)

belongs to the same player as v_0 (resp. v_1) and so on. Given the way the objectives are defined, in the case of reachability or parity SP games, it holds that $\mathsf{pay}(\rho) = \mathsf{pay}(\rho')$ and $\mathsf{won}(\rho) = \mathsf{won}(\rho')$. Therefore, a strategy σ_0 that is solution to the SPS problem in G can be transformed into a strategy $\bar{\sigma}_0$ which is a solution in $\bar{G}$ and vice-versa. ◀