Nash Equilibria in Concurrent Games with Büchi Objectives

Patricia Bouyer, Romain Brenguier, Nicolas Markey, Michael Ummels

March 2015

Research report LSV-15-02 **(Version 1)**

Laboratoire Spécification & Vérification

École Normale Supérieure de Cachan
61, avenue du Président Wilson
94235 Cachan Cedex France

Nash Equilibria in Concurrent Games with Büchi Objectives

Patricia Bouyer, Romain Brenguier, Nicolas Markey, and Michael Ummels

LSV, CNRS & ENS Cachan, France
{bouyer,brenguier,markey,ummels}@lsv.ens-cachan.fr

Abstract

We study the problem of computing pure-strategy Nash equilibria in multiplayer concurrent games with Büchi-definable objectives. First, when the objectives are Büchi conditions on the game, we prove that the existence problem can be solved in polynomial time. In a second part, we extend our technique to objectives defined by deterministic Büchi automata, and prove that the problem then becomes EXPTIME-complete. We prove PSPACE-completeness for the case where the Büchi automata are 1-weak.

1998 ACM Subject Classification F.1.1; F.1.2; F.2.2

Keywords and phrases Concurrent games, Nash equilibria, Büchi objectives

1 Introduction

Game theory (especially games played on graphs) is used in computer science as a powerful framework for modelling interactions in embedded systems [18, 13]. Until recently, more focus had been put on purely antagonistic situations, where the system should fulfil its specification however the environment behaves. This situation can be modelled as a two-player game (one player for the system, and one for the environment), and a winning strategy for the first player is a good controller for the system. In this purely antagonistic view, the objectives of both players are opposite: the aim of the second player is to prevent the first player from achieving her own objective; such games are called zero-sum.

In many cases, however, games are non-zero-sum, especially when they involve more than two players. Such games appear e.g. in various problems in telecommunications, where several agents try to send data on a network [10]. Focusing only on winning strategies in this setting may then be too narrow: winning strategies must be winning against any behaviour of the other agents, and do not consider the fact that the other agents also have their own objectives. In the non-zero-sum setting, each player can have a different payoff associated with an outcome of the game; it is then more interesting to look for *equilibria*. For instance, a Nash equilibrium is a behaviour of the agents in which they play rationally, in the sense that no agent can get a better payoff by unilaterally switching to another strategy [15]. This corresponds to stable states of the game. Note that Nash equilibria need not exist and are not necessarily *optimal*: several equilibria can coexist, possibly with different payoffs.

Our contribution. We focus here on qualitative objectives for the players: such objectives are ω-regular properties over infinite plays, and a player receives payoff 1 if the property is fulfilled and 0 otherwise. Our aim is to decide the existence of pure-strategy Nash equilibria in nondeterministic concurrent games. Being concurrent (instead of the more classical *turn-based* games) and nondeterministic are two important properties of *timed games* (which are games played on timed automata [2, 9]), to which we ultimately want to apply our algorithms.

Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

In a first part, we focus on internal Büchi conditions (defined on the game directly) and show that we can decide the existence of equilibria in polynomial time, which has to be compared with the NP-completeness of the problem in the case of reachability objectives [8, 3]. This relies on an iterated version of a *repellor operator* [3]. Roughly speaking, the repellor is to the computation of Nash equilibria in non-zero-sum games what the attractor is to the computation of winning states in zero-sum games. The repellor operator we use for Büchi objectives is a generalisation of the one defined in [3] for reachability objectives, and the proof techniques are more involved.

Then, using a simulation lemma, we show how to compute Nash equilibria in case the objectives of the players are given by deterministic Büchi automata. This encompasses many winning conditions (among which reachability, Büchi, safety, etc.), and we show that deciding the existence of Nash equilibria with constraints on the payoff is EXPTIME-complete. Under a certain restriction on the automata (1-weakness), we prove that the complexity reduces to PSPACE, and we prove PSPACE-hardness for the special case of safety objectives. When the game is *deterministic* and the 1-weak Büchi automata defining the winning conditions have bounded size (this includes safety and reachability objectives), we show that the constrained existence problem becomes NP-complete. The simulation lemma can also be used to lift our results to timed games, for which all our problems are EXPTIME-complete.

Related work. Concurrent and, more generally, stochastic games go back to Shapley [17]. However, most research in game theory and economics has focused on games with rewards, which are either averaged or discounted along an infinite path. In particular, Fink [11] proved that every discounted stochastic game has a Nash equilibrium in pure strategies, and Vielle [22] proved the existence of *ϵ-equilibria* in randomised strategies for two-player stochastic games under the average-reward criterion. For two-player concurrent games with Büchi objectives, the existence of ϵ-equilibria (in randomised strategies) was proved by Chatterjee [5]. However, exact Nash equilibria need not exist. An important subclass where even Nash equilibria in pure strategies exist are turn-based games with Büchi objectives [8].

The complexity of Nash equilibria in games played on graphs was first addressed in [8, 19]. In particular, it was shown in [19] that the existence of a Nash equilibrium with a constraint on its payoff can be decided in polynomial time for turn-based games with Büchi objectives. In this paper, we extend this result to concurrent games. It was also shown in [19] that the same problem is NP-hard for turn-based games with co-Büchi conditions, which implies hardness for concurrent games with this kind of objectives. For concurrent games with ω-regular objectives, the decidability of the constrained existence problem w.r.t. pure strategies was established by Fisman et al. [12], but their algorithm runs in doubly exponential time, whereas our algorithm for Büchi games runs in polynomial time. Finally, Ummels and Wojtczak [21] proved that the existence of a Nash equilibrium in pure or randomised strategies is undecidable for *stochastic* games with reachability or Büchi objectives, which justifies our restriction to concurrent games without probabilistic transitions (see [20] for a similar undecidability result for randomised Nash equilibria in non-stochastic games).

2 Preliminaries

2.1 Concurrent Games

A *transition system* is a 2-tuple $\mathcal{S} = \langle \text{States}, \text{Edg} \rangle$ where States is a (possibly uncountable) set of states and $\text{Edg} \subseteq \text{States} \times \text{States}$ is the set of transitions. In a transition system $\mathcal{S}$, a *path* π is a non-empty sequence $(s_i)_{0 \le i < n}$ (where $n \in \mathbb{N} \cup \{\infty\}$) of states such that $(s_i, s_{i+1}) \in \text{Edg}$

for all $i < n-1$. The *length* of π, denoted by $|\pi|$, is $n-1$. The set of finite paths (also called *histories*) of $\mathcal{S}$ is denoted by $\mathrm{Hist}_{\mathcal{S}}$, the set of infinite paths (also called *plays*) of $\mathcal{S}$ is denoted by $\mathrm{Play}_{\mathcal{S}}$, and $\mathrm{Path}_{\mathcal{S}} = \mathrm{Hist}_{\mathcal{S}} \cup \mathrm{Play}_{\mathcal{S}}$ is the set of paths of $\mathcal{S}$. Given a path $\pi = (s_i)_{0 \le i < n}$ and an integer $j < n$, the *j-th prefix* (resp. *j-th suffix*, *j-th state*) of π, denoted by $\pi_{\le j}$ (resp. $\pi_{\ge j}$, $\pi_{=j}$), is the finite path $(s_i)_{0 \le i < j+1}$ (resp. $(s_i)_{j \le i < n}$, state s_j). If $\pi = (s_i)_{0 \le i < n}$ is a history, we write $\mathrm{last}(\pi) = s_{|\pi|}$.

We consider nondeterministic concurrent games [3], which extend standard concurrent games [1] with nondeterminism.

▶ **Definition 1.** A *(nondeterministic) concurrent game* is a tuple $\mathcal{G} = \langle \mathrm{States}, \mathrm{Edg}, \mathrm{Agt}, \mathrm{Act}, \mathrm{Mov}, \mathrm{Tab}, (\mathcal{L}_A)_{A \in \mathrm{Agt}} \rangle$, where $\langle \mathrm{States}, \mathrm{Edg} \rangle$ is a transition system, Agt is a finite set of players, Act is a (possibly uncountable) set of actions, and

- $\mathrm{Mov}\colon \mathrm{States} \times \mathrm{Agt} \to 2^{\mathrm{Act}} \setminus \{\varnothing\}$ is a mapping indicating the actions available to a given player in a given state;
- $\mathrm{Tab}\colon \mathrm{States} \times \mathrm{Act}^{\mathrm{Agt}} \to 2^{\mathrm{Edg}} \setminus \{\varnothing\}$ associates with a state and an action profile the resulting set of edges; we require that $s = s'$ if $(s', s'') \in \mathrm{Tab}(s, \langle m_A \rangle_{A \in \mathrm{Agt}})$;
- $\mathcal{L}_A \subseteq \mathrm{States}^\omega$ defines the *objective* for player $A \in \mathrm{Agt}$; the *payoff* for player A is the function $\nu_A\colon \mathrm{States}^\omega \to \{0,1\}$, where $\nu_A(\pi) = 1$ if $\pi \in \mathcal{L}_A$, and $\nu_A(\pi) = 0$ otherwise; we say that player A *prefers* play π' over play π, denoted $\pi \preccurlyeq_A \pi'$, if $\nu_A(\pi) \le \nu_A(\pi')$.

We call a game $\mathcal{G}$ *finite* if its set of states is finite.

Non-determinism naturally appears in timed games, and this is our most important motivation for investigating this extension of standard concurrent games. We explain in Section 4.3 how our results apply to timed games.

We say that a *move* $\langle m_A \rangle_{A \in \mathrm{Agt}} \in \mathrm{Act}^{\mathrm{Agt}}$ (which we may write m_{Agt} in the sequel) is *legal* at s if $m_A \in \mathrm{Mov}(s, A)$ for all $A \in \mathrm{Agt}$. A concurrent game is *deterministic* if $\mathrm{Tab}(s, m_{\mathrm{Agt}})$ is a singleton for each $s \in \mathrm{States}$ and each legal move m_{Agt} (at s). A game is *turn-based* if for each state the set of allowed moves is a singleton for all but at most one player.

In a nondeterministic concurrent game, whenever we arrive at a state s, the players (simultaneously) choose a legal move m_{Agt}. Then, one of the transitions in $\mathrm{Tab}(s, m_{\mathrm{Agt}})$ is nondeterministically selected, which results in a new state of the game. In the sequel, we write $\mathrm{Hist}_{\mathcal{G}}$, $\mathrm{Play}_{\mathcal{G}}$ and $\mathrm{Path}_{\mathcal{G}}$ for the corresponding set of paths in the underlying transition system of $\mathcal{G}$. We also write $\mathrm{Hist}_{\mathcal{G}}(s)$, $\mathrm{Play}_{\mathcal{G}}(s)$ and $\mathrm{Path}_{\mathcal{G}}(s)$ for the respective subsets of paths starting in state s.

▶ **Definition 2.** Let $\mathcal{G}$ be a concurrent game, and $A \in \mathrm{Agt}$. A *strategy* for A is a mapping $\sigma_A\colon \mathrm{Hist}_{\mathcal{G}} \to \mathrm{Act}$ such that $\sigma_A(\pi) \in \mathrm{Mov}(\mathrm{last}(\pi), A)$ for all $\pi \in \mathrm{Hist}_{\mathcal{G}}$. A strategy σ_P for a coalition P is a tuple of strategies, one for each player in P. We write $\sigma_P = \langle \sigma_A \rangle_{A \in P}$ for such a strategy. A *strategy profile* is a strategy for the coalition Agt. We write $\mathrm{Strat}^P_{\mathcal{G}}$ for the set of strategies of coalition P (or simply $\mathrm{Strat}^B_{\mathcal{G}}$ if $P = \{B\}$), and $\mathrm{Prof}_{\mathcal{G}} = \mathrm{Strat}^{\mathrm{Agt}}_{\mathcal{G}}$.

Note that we only consider non-randomised (*pure*) strategies in this paper. Notice also that strategies are based on the sequences of visited states, and not on the actions played by the players. This is a realistic assumption for concurrent systems where different components interact with each other and each component has a set of internal actions which cannot be observed by the other components. However, it makes the computation of equilibria harder: when a deviation from the equilibrium profile occurs, the given sequence of states does not uniquely determine the player who has deviated. While the main part of the paper focuses on state-based strategies for Büchi objectives, we show in Section 6 that equilibria in the action-based setting can be computed more easily, even for parity objectives.

Let $\mathcal{G}$ be a game, P a coalition, and σ_P a strategy for P. A path $\pi = (s_j)_{0 \leq j \leq |\pi|}$ is *compatible* with the strategy σ_P if, for all $k < |\pi|$, there exists a move m_{Agt} such that (i) m_{Agt} is legal at s_k, (ii) $m_A = \sigma_A(\pi_{\leq k})$ for all $A \in P$, and (iii). $(s_k, s_{k+1}) \in \mathrm{Tab}(s_k, m_{\mathrm{Agt}})$. We write $\mathrm{Out}_{\mathcal{G}}(\sigma_P)$ for the set of paths (or *outcomes*) in $\mathcal{G}$ that are compatible with the strategy σ_P, and we write $\mathrm{Out}^{\mathrm{f}}_{\mathcal{G}}(\sigma_P)$ (resp. $\mathrm{Out}^{\infty}_{\mathcal{G}}(\sigma_P)$) for the finite (resp. infinite) outcomes, and $\mathrm{Out}_{\mathcal{G}}(s, \sigma_P)$, $\mathrm{Out}^{\mathrm{f}}_{\mathcal{G}}(s, \sigma_P)$ and $\mathrm{Out}^{\infty}_{\mathcal{G}}(s, \sigma_P)$ for the respective sets of outcomes that start in state s. In general there might be several infinite outcomes for a strategy profile from a given state. However, in the case of deterministic games, any strategy profile has a single infinite outcome from a given state.

▶ **Example 3.** Figure 1 depicts a two-player concurrent game, called the *matching-penny* game. A pair $\langle a, b\rangle$ represents a move, where Player 1 plays action a and Player 2 plays b. Starting from state s_0, if both players choose the same action, then the game proceeds to s_1; otherwise, the game proceeds to s_2. In the matching-penny game, the objective for Player 1 is to visit s_1 (which is encoded as $\mathcal{L}_1 = \mathrm{States}^* \cdot \{s_1\} \cdot \mathrm{States}^\omega$), while for Player 2 it is to visit s_2. Figure 2 shows another game (our running example), in which the objective of Player 1 is to *loop* in s_1 ($\mathcal{L}_1 = \mathrm{States}^* \cdot \{s_1\}^\omega$), whereas the objective for Player 2 is to *loop* in s_2 ($\mathcal{L}_2 = \mathrm{States}^* \cdot \{s_2\}^\omega$).

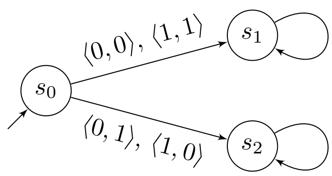

Figure 1 The matching-penny game

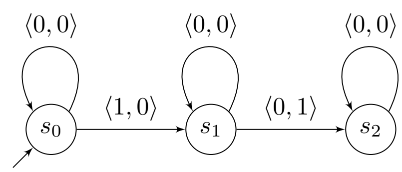

Figure 2 A game with Büchi objectives

2.2 Pseudo-Nash Equilibria

Given a move m_{Agt} and an action m' for some player B, we write $m_{\mathrm{Agt}}[B \mapsto m']$ for the move n_{Agt} with $n_A = m_A$ if $A \neq B$ and $n_B = m'$. This is extended to strategies in the natural way. For non-zero-sum games, several notions of equilibria have been defined, e.g. Nash equilibria [15], subgame-perfect equilibria [16], and secure equilibria [6]. None of these notions apply to nondeterministic games. Bouyer et al. have therefore proposed the notion of *pseudo-Nash equilibria* [3], which extend standard Nash equilibria to nondeterministic games.

▶ **Definition 4.** Let $\mathcal{G}$ be a nondeterministic concurrent game with objectives $(\mathcal{L}_A)_{A \in \mathrm{Agt}}$, and let s be a state of $\mathcal{G}$. A *pseudo-Nash equilibrium* of $\mathcal{G}$ from s is a pair $(\sigma_{\mathrm{Agt}}, \pi)$ of a strategy profile $\sigma_{\mathrm{Agt}} \in \mathrm{Prof}_{\mathcal{G}}$ and a play $\pi \in \mathrm{Out}_{\mathcal{G}}(s, \sigma_{\mathrm{Agt}})$ such that $\pi' \preccurlyeq_B \pi$ for all players $B \in \mathrm{Agt}$, all strategies $\sigma' \in \mathrm{Strat}^B$, and all plays $\pi' \in \mathrm{Out}_{\mathcal{G}}(s, \sigma_{\mathrm{Agt}}[B \mapsto \sigma'])$. The outcome π is then called an *optimal play* for the strategy profile σ_{Agt}.

For deterministic games, the play π is uniquely determined by σ_{Agt}, so that pseudo-Nash equilibria coincide with *Nash equilibria* [15]: these are strategy profiles where no player has an incentive to unilaterally deviate from her strategy.

In the case of nondeterministic games, a strategy profile for an equilibrium may give rise to several outcomes. The outcome π is then chosen cooperatively by all players: once a strategy profile is fixed, nondeterminism is resolved by all players choosing one of the possible outcomes in such a way that each player has no incentive to unilaterally changing her choice (nor her strategy). To the best of our knowledge, this cannot be encoded by adding an extra player for resolving the nondeterminism.

▶ **Example 5.** Clearly, there is no pure-strategy Nash equilibrium in the game of Figure 1 since a losing player can always improve her payoff by switching her choice. In the game of Figure 2 (where Player i wants to visit s_i infinitely often), there are several Nash equilibria: one with payoff $(0, 1)$, in which both players play action 1 when it is available; another one with payoff $(0, 0)$, in which Player 1 always plays 0, and Player 2 plays 1 when available.

In this paper, we study several decision problems related to the existence of pseudo-Nash equilibria. The *existence problem* consists in deciding the existence of a pseudo-Nash equilibrium in a given state of a game. Since several pseudo-Nash equilibria may coexist, it is also interesting to decide whether there is one with a given payoff (0 or 1) for some of the players; this is the *constrained existence problem*. Finally, the *verification problem* asks whether a given payoff function (totally defined over Agt) is the payoff of some pseudo-Nash equilibrium. Notice that the first and third problems are trivially logspace-reducible to the second one.

3 Internal Büchi Objectives

In this section, we fix a nondeterministic concurrent game $\mathcal{G} = \langle \mathrm{States}, \mathrm{Edg}, \mathrm{Agt}, \mathrm{Act}, \mathrm{Mov}, \mathrm{Tab}, (\mathcal{L}_A)_{A\in\mathrm{Agt}}\rangle$, where the objectives are *internal* Büchi conditions given by a set $\Omega_A \subseteq \mathrm{States}$ of target states for each player $A \in \mathrm{Agt}$. The corresponding objective for player A is the set $\mathcal{L}_A = \{\pi \in \mathrm{States}^\omega \mid \pi_{=j} \in \Omega_A \text{ for infinitely many } j \in \mathbb{N}\}$.

3.1 Characterising Equilibria Using Fixpoints

In [3], pseudo-Nash equilibria are characterised for qualitative reachability objectives using a fixpoint computation called the *repellor*. This was the counter-part of the attractor in non-zero-sum games for computing equilibria. In this section, we extend the repellor to handle internal Büchi objectives.

Suspect players. Let $e = (s, s')$ be an edge. Given a move m_{Agt}, we define the set of suspect players for e as the set

$$\mathrm{Susp}(e, m_{\mathrm{Agt}}) = \{B \in \mathrm{Agt} \mid \exists\, m' \in \mathrm{Mov}(s, B) \text{ such that } e \in \mathrm{Tab}(s, m_{\mathrm{Agt}}[B \mapsto m'])\}.$$

Intuitively, Player $B \in \mathrm{Agt}$ is a suspect for edge e and if she can unilaterally change her action to trigger edge e. Notice that if $e \in \mathrm{Tab}(s, m_{\mathrm{Agt}})$, then $\mathrm{Susp}(e, m_{\mathrm{Agt}}) = \mathrm{Agt}$.

The iterated (or Büchi) repellor. For any $n \in \mathbb{N}$ and $P \subseteq \mathrm{Agt}$, we define the *n-th repellor set* $\mathrm{Rep}^n_{\mathcal{G}}(P)$ as follows. If $n = 0$, then $\mathrm{Rep}^0_{\mathcal{G}}(P) = \emptyset$ for any $P \subseteq \mathrm{Agt}$. Now fix $n \in \mathbb{N}$, and assume that repellor sets $\mathrm{Rep}^n_{\mathcal{G}}(P)$ have been defined for any $P \subseteq \mathrm{Agt}$. As the base case for level $n+1$, we set $\mathrm{Rep}^{n+1}_{\mathcal{G}}(\emptyset) = \mathrm{States}$. Then, assuming that $\mathrm{Rep}^{n+1}_{\mathcal{G}}(P')$ has been defined for all $P' \subsetneq P$, we let $\mathrm{Rep}^{n+1}_{\mathcal{G}}(P)$ be the largest set fulfilling the following condition: for all $s \in \mathrm{Rep}^{n+1}_{\mathcal{G}}(P)$ there exists a legal move m_{Agt} (at s) such that

1. $s' \in \mathrm{Rep}^{n+1}_{\mathcal{G}}(P \cap \mathrm{Susp}_{\mathcal{G}}((s, s'), m_{\mathrm{Agt}}))$ for all $s' \in \mathrm{States}$, and
2. if $s' \in \Omega_A$ for some player $A \in P \cap \mathrm{Susp}_{\mathcal{G}}((s, s'), m_{\mathrm{Agt}})$, then $s' \in \mathrm{Rep}^n_{\mathcal{G}}(P \cap \mathrm{Susp}_{\mathcal{G}}((s, s'), m_{\mathrm{Agt}}))$.

Given a state $s \in \mathrm{Rep}^{n+1}_{\mathcal{G}}(P)$, a legal move m_{Agt} that fulfils 1. and 2. is called a *secure move* (w.r.t. P and $n+1$); we write $\mathrm{Secure}^{n+1}_{\mathcal{G}}(s, P)$ for the set of these moves. Finally, we define $\mathrm{Rep}^{\infty}_{\mathcal{G}}(P) = \bigcup_{n \geq 0} \mathrm{Rep}^n(P)$. In the following, to improve readability, we will omit the index $\mathcal{G}$ in all the notions we have defined, when the game is clear from the context.

Intuitively, a state s is an element of $\mathrm{Rep}^n_{\mathcal{G}}(P)$ iff at s there is a legal move such that no player $A \in P$ can force to visit her set of target states at least n times by changing her action. For finite games, it follows that a state s is an element of $\mathrm{Rep}^{\infty}_{\mathcal{G}}(P)$ iff at s there is a legal move such that no player $A \in P$ can force to visit her set set of target states *infinitely often* by changing her action.

▶ Remark. The repellor defined for reachability objectives in [3] is rather similar to $\mathrm{Rep}^1(P)$; it differs only in the second condition, which was "$\mathrm{Rep}^1(P) \cap \Omega_A = \emptyset$ for all $A \in P$" in [3]. This change is required since a play that is losing w.r.t. a Büchi objective might visit a winning state a finite number of times (as opposed to reachability objectives, where this cannot happen).

▶ **Example 6.** In the game of Figure 2, if we assume reachability objectives (state s_i for Player i), there is no equilibrium with payoff $(0, 0)$, since Player 1 can enforce a visit to her winning state. If we assume Büchi objectives, we have seen in Example 5 that there is an equilibrium with payoff $(0, 0)$. Table 1 displays the values of the iterated repellors in this game, for all possible sets of players. These results were obtained with our prototype implementation of our algorithms, available at `http://www.lsv.ens-cachan.fr/Software/praline/`.

Table 1 Computing the repellor sets in the game of Figure 2

P	$\mathrm{Rep}^0(P)$	$\mathrm{Rep}^1(P)$	$\mathrm{Rep}^2(P)$	$\mathrm{Rep}^{\infty}(P) = \mathrm{Rep}^3(P)$
$\emptyset$	$\emptyset$	$\{s_0, s_1, s_2\}$	$\{s_0, s_1, s_2\}$	$\{s_0, s_1, s_2\}$
$\{A_1\}$	$\emptyset$	$\{s_1, s_2\}$	$\{s_0, s_1, s_2\}$	$\{s_0, s_1, s_2\}$
$\{A_2\}$	$\emptyset$	$\{s_0\}$	$\{s_0\}$	$\{s_0\}$
$\{A_1, A_2\}$	$\emptyset$	$\emptyset$	$\{s_0\}$	$\{s_0\}$

▶ **Lemma 7.** *The repellor and the secure moves satisfy the following properties:*

- *If* $P' \subseteq P \subseteq \mathrm{Agt}$, *then* $\mathrm{Rep}^n(P) \subseteq \mathrm{Rep}^n(P')$ *for all* $n \in \mathbb{N}$.
- $\mathrm{Rep}^n(P) \subseteq \mathrm{Rep}^{n+1}(P)$ *for all* $P \subseteq \mathrm{Agt}$ *and* $n \in \mathbb{N}$.
- $\mathrm{Secure}^n(s, P) \subseteq \mathrm{Secure}^{n+1}(s, P)$ *for all* $P \subseteq \mathrm{Agt}$, $n \in \mathbb{N}$ *and* $s \in \mathrm{States}$.

We define the *n-th repellor transition system* $\mathcal{S}^n(P) = \langle \mathrm{States}, \mathrm{Edg}_n \rangle$ by $(s, s') \in \mathrm{Edg}_n$ iff there exists $m_{\mathrm{Agt}} \in \mathrm{Secure}^n(s, P)$ such that $(s, s') \in \mathrm{Tab}(s, m_{\mathrm{Agt}})$. Note in particular that any $s \in \mathrm{Rep}^n(P)$ has an outgoing transition in $\mathcal{S}^n(P)$. We also define the limit repellor transition system $\mathcal{S}^{\infty}(P) = \langle \mathrm{States}, \bigcup_{n \geq 0} \mathrm{Edg}_n \rangle$. The following lemma bounds the number of iteration steps required to reach $\mathrm{Rep}^{\infty}(P)$.

▶ **Lemma 8.** *Let $\mathcal{G}$ be a a finite game, $P \subseteq \mathrm{Agt}$, and let ℓ be the length of the longest acyclic path in $\mathcal{G}$. Then $\mathrm{Rep}^n(P) = \mathrm{Rep}^{\infty}(P)$ for all $n \geq \ell \cdot |P|$.*

The correctness of the iterated repellor for *finite* games is stated in the next proposition, whose proof can be found in the appendix.

▶ **Proposition 9.** *Let $\mathcal{G}$ be a finite game, $P \subseteq \text{Agt}$, and let $\rho \in \text{Play}(s)$ be a play that visits $\bigcup_{B \in P} \Omega_B$ only finitely often. Then ρ is a path in $\mathcal{S}^\infty(P)$ if and only if there exists $\sigma_{\text{Agt}} \in \text{Prof}$ such that $\rho \in \text{Out}(s, \sigma_{\text{Agt}})$ and ρ' does not visit Ω_B infinitely often for all plays ρ' that can arise when some player $B \in P$ changes her strategy, i.e. $\rho' \in \text{Out}(s, \sigma_{\text{Agt}}[B \mapsto \sigma'])$ for some $B \in P$ and some $\sigma' \in \text{Strat}^B$.*

We can deduce from this proposition that if ρ is an infinite path from state s in $\mathcal{S}^\infty(P)$ that visits Ω_A infinitely often *if and only if* $A \notin P$, then there is a pseudo Nash equilibrium from s with optimal play ρ.

▶ **Corollary 10.** *Let $\mathcal{G}$ be a finite game, $s \in \text{States}$, and $\nu \colon \text{Agt} \to \{0, 1\}$. There exists a pseudo-Nash equilibrium in $\mathcal{G}$ with payoff ν if and only if there exists an infinite path ρ in $\mathcal{S}^\infty(\nu^{-1}(0))$ with payoff $\nu_A(\rho) = \nu(A)$ for all $A \in \text{Agt}$.*

3.2 Application to Solving the Three Problems

We use the previous characterisation for analysing the complexity of the various decision problems that we have defined in Section 2.2.

▶ **Theorem 11.** *The verification, existence and constrained existence problems for finite games with internal Büchi objectives are* PTIME*-complete.*

The lower bounds are simple adaptations of the PTIME-hardness of the circuit value problem (see Appendix A.3). We now focus on the PTIME upper bounds, and prove it for the constrained existence problem (which implies the same upper bound for the other two problems).

We first use the equivalence given in Proposition 9 to get a set-based characterisation of (pseudo-)Nash equilibria. We fix a set of winning players $W \subseteq \text{Agt}$ and a set of losing players $L \subseteq \text{Agt}$, and we fix an initial state s. Given a transition system $\langle S, E \rangle$ and a set of players P, we say that they satisfy condition (‡) if the following properties are fulfilled:

(1) $\Omega_A \cap S = \emptyset$ if and only if $A \in P$;
(2) $L \subseteq P$ and $P \cap W = \emptyset$;
(3) $\langle S, E \rangle$ is strongly connected;
(4) $\langle S, E \rangle \subseteq \mathcal{S}^\infty(P)$;
(5) S is reachable from s in $\mathcal{S}^\infty(P)$.

The following is then a corollary to Proposition 9.

▶ **Corollary 12** (Set-based characterisation)**.** *A pair $(\langle S, E \rangle, P)$ satisfies condition (‡) iff there is an infinite path ρ in $\mathcal{S}^\infty(P)$ from s that, from some point onwards, stays in $\langle S, E \rangle$. In particular, ρ is losing for all players in L. Moreover, if $(\langle S, E \rangle, P)$ satisfies (‡), then there exists an infinite path ρ in $\mathcal{S}^\infty(P)$ from s with the same property that visits* all *states of S infinitely often (and is thus winning for all players in W).*

Note that in the above characterisation, P is uniquely determined by the set S; hence we write $P(S) = \{A \in \text{Agt} \mid S \cap \Omega_A = \emptyset\}$, and we say that $\langle S, E \rangle$ satisfies condition (‡) if $(\langle S, E \rangle, P(S))$ does. Our aim is to compute in polynomial time all *maximal* pairs $\langle S, E \rangle$ that satisfy condition (‡). As a prerequisite, we assume that we can compute $\mathcal{S}^\infty(P)$ in polynomial time whenever $P \subseteq \text{Agt}$ is given. This can be proved using similar arguments as in [4] (see Appendix A.4). Now, we define a recursive operator SSG (SolveSubGame) by setting $\mathsf{SSG}(\langle S, E \rangle) = \{\langle S, E \rangle\}$ if $\langle S, E \rangle \subseteq \mathcal{S}^\infty(P(S))$ and $\langle S, E \rangle$ is strongly connected, and

$$\mathsf{SSG}(\langle S, E \rangle) = \bigcup_{\langle S', E' \rangle \in \text{SCC}(\langle S, E \rangle)} \mathsf{SSG}(\langle S', E' \rangle \cap \mathcal{S}^\infty(P(S')))$$

in all other cases. Here, $\mathrm{SCC}(\langle S, E\rangle)$ denotes the set of strongly connected components of $\langle S, E\rangle$ (which can be computed in linear time). Finally, we define

$$\mathsf{Sol}(L, W) = \mathsf{SSG}\Big(\langle \mathrm{States} \setminus \bigcup_{A\in L} \Omega_A, \mathrm{Edg}\rangle\Big) \cap \{\langle S, E\rangle \mid P(S) \cap W = \emptyset\}\,.$$

▶ **Lemma 13.** *If $\langle S, E\rangle \in \mathsf{Sol}(L, W)$ and S is reachable from s in $\mathcal{S}^\infty(P(S))$, then it satisfies condition (‡). Conversely, if $\langle S, E\rangle$ satisfies condition (‡), then there exists $\langle S', E'\rangle \in \mathsf{Sol}(L, W)$ such that $\langle S, E\rangle \subseteq \langle S', E'\rangle$.*

▶ **Lemma 14.** *The set $\mathsf{Sol}(L, W)$ can be computed in polynomial time.*

The PTIME upper bound of Theorem 11 follows from the above analysis (see Appendix A.7).

▶ Remark. This result may seem surprising since we know that the problem is NP-complete for reachability objectives, even in turn-based games [8, 3]. Intuitively, the problem is harder for reachability objectives because whether a play satisfies or not a reachability objective is not only determined by its behaviour in the strongly connected component in which it settles but on *all* visited states.

4 Game Simulations

Our aim is to transfer our results for internal Büchi objectives to larger classes of objectives. A useful tool is the notion of game simulation, which we develop now.

4.1 Definition and General Properties

We already gave a definition of game simulation in [3], which was tailored to games with reachability objectives; we extend this notion to games with arbitrary qualitative objectives.

▶ **Definition 15.** Consider two games $\mathcal{G} = \langle \mathrm{States}, \mathrm{Edg}, \mathrm{Agt}, \mathrm{Act}, \mathrm{Mov}, \mathrm{Tab}, (\mathcal{L}_A)_{A\in\mathrm{Agt}}\rangle$ and $\mathcal{G}' = \langle \mathrm{States}', \mathrm{Edg}', \mathrm{Agt}, \mathrm{Act}', \mathrm{Mov}', \mathrm{Tab}', (\mathcal{L}'_A)_{A\in\mathrm{Agt}}\rangle$ with the same set Agt of players. A relation $\lhd \subseteq \mathrm{States} \times \mathrm{States}'$ is a *game simulation* if $s \lhd s'$ implies that for each move m_{Agt} in $\mathcal{G}$ there exists a move m'_{Agt} in $\mathcal{G}'$ such that

1. for each $t' \in \mathrm{States}'$ there exists $t \in \mathrm{States}$ with $t \lhd t'$ and $\mathrm{Susp}((s', t'), m'_{\mathrm{Agt}}) \subseteq \mathrm{Susp}((s, t), m_{\mathrm{Agt}})$, and
2. for each $(s, t) \in \mathrm{Tab}(s, m_{\mathrm{Agt}})$ there exists $(s', t') \in \mathrm{Tab}'(s', m'_{\mathrm{Agt}})$ with $t \lhd t'$.

If $\lhd$ is a game simulation, we say that $\mathcal{G}'$ *simulates* $\mathcal{G}$. Finally, a game simulation $\lhd$ is *winning-preserving* from $(s_0, s'_0) \in \mathrm{States}\times\mathrm{States}'$ if for all $\rho \in \mathrm{Play}_{\mathcal{G}}(s_0)$ and $\rho' \in \mathrm{Play}_{\mathcal{G}'}(s'_0)$ with $\rho \lhd \rho'$ (i.e. $\rho_{=i} \lhd \rho'_{=i}$ for all $i \in \mathbb{N}$) it holds that $\rho \in \mathcal{L}_A$ iff $\rho' \in \mathcal{L}'_A$ for all $A \in \mathrm{Agt}$.

▶ **Proposition 16.** *Game simulation is transitive.*

▶ **Proposition 17.** *Assume $\mathcal{G}$ and $\mathcal{G}'$ are games. Fix two states s and s' in $\mathcal{G}$ and $\mathcal{G}'$ respectively, and assume that $\lhd$ is a winning-preserving game simulation from (s, s'). If there exists a pseudo-Nash equilibrium $(\sigma_{\mathrm{Agt}}, \rho)$ of $\mathcal{G}$ from s, then there exists a pseudo-Nash equilibrium $(\sigma'_{\mathrm{Agt}}, \rho')$ of $\mathcal{G}'$ from s' with $\rho \lhd \rho'$. In particular, ρ and ρ' have the same payoff.*

4.2 Product of a Game with Deterministic Büchi Automata

We use the results on game simulation to study objectives that are defined by deterministic Büchi automata. A *deterministic Büchi automaton* $\mathcal{A}$ over Σ is a tuple $\langle Q, \Sigma, \delta, q_0, R\rangle$, where Q is a finite set of *states*, Σ is the *input alphabet*, $\delta\colon Q \times \Sigma \to Q$ is the *transition function*, $q_0 \in Q$ is the *initial* state, and $R \subseteq Q$ is the set of *repeated* states. We assume the reader is familiar with Büchi automata, and we write $\mathrm{L}(\mathcal{A}) \subseteq \Sigma^\omega$ for the language accepted by $\mathcal{A}$.

Fix a game $\mathcal{G} = \langle \mathrm{States}, \mathrm{Edg}, \mathrm{Agt}, \mathrm{Act}, \mathrm{Mov}, \mathrm{Tab}, (\mathcal{L}_B)_{B\in\mathrm{Agt}}\rangle$ and a player $A \in \mathrm{Agt}$, and assume that $\mathcal{L}_A = \mathrm{L}(\mathcal{A})$ for a deterministic Büchi automaton $\mathcal{A} = \langle Q, \mathrm{States}, \delta, q_0, R\rangle$ over States. We show how to compute pseudo-Nash equilibria in $\mathcal{G}$ by building a product of $\mathcal{G}$ with $\mathcal{A}$ and computing the pseudo-Nash equilibria in the resulting game.

We define the product of the game $\mathcal{G}$ with the automaton $\mathcal{A}$ as the game $\mathcal{G} \ltimes \mathcal{A} = \langle \mathrm{States}', \mathrm{Edg}', \mathrm{Agt}, \mathrm{Act}, \mathrm{Mov}', \mathrm{Tab}', (\mathcal{L}'_B)_{B\in\mathrm{Agt}}\rangle$, where:

- $\mathrm{States}' = \mathrm{States} \times Q$;
- $\mathrm{Edg}' = \{((s,q),(s',q')) \mid (s,s') \in \mathrm{Edg} \text{ and } \delta(q,s) = q'\}$;
- $\mathrm{Mov}'((s,q), A_i) = \mathrm{Mov}(s, A_i)$ for every $A_i \in \mathrm{Agt}$;
- $\mathrm{Tab}'((s,q), m_{\mathrm{Agt}}) = \{((s,q),(s',q')) \mid (s,s') \in \mathrm{Tab}(s, m_{\mathrm{Agt}}) \text{ and } \delta(q,s) = q'\}$;
- if $B = A$, then $\mathcal{L}'_B$ is the internal Büchi objective given by the set $\Omega = \mathrm{States} \times R$; otherwise, $\mathcal{L}'_B = \pi^{-1}(\mathcal{L}_B)$, where π is the natural projection from States' to States and its extension to plays (i.e. $\pi((s_0,q_0)(s_1,q_1)\dots) = s_0 s_1 \dots$).

▶ Remark. Note that, if $\mathcal{L}_B$ is defined by an internal Büchi condition, then so is $\mathcal{L}'_B$.

▶ **Lemma 18.** *$\mathcal{G} \ltimes \mathcal{A}$ simulates $\mathcal{G}$, and vice versa. Furthermore, in both cases we can exhibit a game-simulation that is winning-preserving from $(s,(s,q_0))$ for all $s \in \mathrm{States}$.*

Assume that for each player $A_i \in \mathrm{Agt}$ the objective in $\mathcal{G}$ is given by a deterministic Büchi automaton $\mathcal{A}_i$. We use the transitivity of game simulation to build a product of $\mathcal{G}$ with each of the automata $\mathcal{A}_i$, namely $\mathcal{G}' = \mathcal{G} \ltimes \mathcal{A}_1 \ltimes \dots \ltimes \mathcal{A}_n$ (we assume that $\ltimes$ is left-associative). Each player $A_i \in \mathrm{Agt}$ has an internal Büchi objective in $\mathcal{G}'$, which we denote by Ω_i.

▶ **Corollary 19.** *Let $s \in \mathrm{States}$ and $\nu\colon \mathrm{Agt} \to \{0,1\}$. There is a pseudo-Nash equilibrium in $\mathcal{G}$ from s with payoff ν if and only if there is a pseudo-Nash equilibrium in $\mathcal{G}'$ from $(s, q_{01}, \dots, q_{0n})$ with payoff ν, where q_{0i} is the initial state of $\mathcal{A}_i$.*

4.3 Application to Timed Games

We now apply the game-simulation approach to the computation of pseudo-Nash equilibria in timed games. Given a timed game $\mathcal{G}$ with internal Büchi objectives, the corresponding (exponential-size) region game $\mathcal{R}_\mathcal{G}$ as defined in [4] simulates $\mathcal{G}$ and is simulated by $\mathcal{G}$ while preserving winning conditions (the proof for reachability objectives in [4] can be easily extended to our framework). Pseudo-Nash equilibria of $\mathcal{G}$ can thus be computed on the finite game $\mathcal{R}_\mathcal{G}$. If the objectives of the players are defined by deterministic Büchi automata $(\mathcal{A}_i)_{A_i\in\mathrm{Agt}}$, we can compute the product $\mathcal{R}_\mathcal{G} \ltimes \mathcal{A}_1 \ltimes \dots \ltimes \mathcal{A}_n$ with corresponding internal Büchi objectives $(\Omega_i)_{A_i\in\mathrm{Agt}}$, as defined in the previous subsection. This product has size exponential in the size of $\mathcal{G}$ and in the number of players. We can then apply the algorithm developed in Section 3.2, yielding an EXPTIME upper bound for deciding the verification, existence, and constrained existence problems in timed games. Finally we get EXPTIME-hardness for internal Büchi objectives by applying the reduction in [4, Prop. 20] (replace all accepting sink states by repeated states).

▶ **Theorem 20.** *The verification, existence, and constrained existence problems are* EXPTIME*-complete both for timed games with internal Büchi objectives and for timed games with objectives defined by deterministic Büchi automata.*

5 Büchi-Definable Objectives

The characterisation of Corollary 19 gives a procedure to compute pseudo-Nash equilibria in games with objectives defined by deterministic Büchi automata (one automaton per player). The algorithm runs in time exponential in the number of players since we have to build the product with all the deterministic Büchi automata defining the objective of a player. We prove that our problems are EXPTIME-hard by encoding two-player countdown games [14] into multiplayer games. Each bit of the countdown will be managed by a different player, who is in charge of checking that this bit is correctly updated at each transition.

▶ **Theorem 21.** *The verification, existence, and constrained existence problems for finite games with objectives defined by deterministic Büchi automata are* EXPTIME*-complete.*

We now prove that when the deterministic Büchi automata defining the objectives are 1-*weak* (i.e. when all strongly connected components of the transition graph consist of just one state), all our three problems can be solved in PSPACE. In particular, this result applies to safety (and reachability) objectives, which can be defined by 1-weak automata. We define a procedure (see Algorithm 1 in Appendix C.2) that, given parameters (P, n, q), computes the set of states s such that in the product game $(s, q) \in \mathrm{Rep}^n(P)$. The procedure computes the repellor as a fixpoint, calling itself recursively on instances (P', n', q'), where either $P' \subsetneq P$, $n' < n$, or q' is a successor of q. The maximal number of nested calls is $|P| + n + \sum_{i \in \mathrm{Agt}} \ell_i$, where ℓ_i is the length of the longest acyclic path in $\mathcal{A}_i$. According to Lemma 8, n can be bounded by a polynomial. The whole computation thus runs in polynomial space.

▶ **Theorem 22.** *The verification, existence, and constrained existence problems are in* PSPACE *for finite games with objectives defined by* 1*-weak deterministic Büchi automata.*

The matching lower bound holds already for the special case of safety objectives.

▶ **Proposition 23.** *The verification, existence, and constrained existence problems are* PSPACE*-hard for finite games with safety objectives.*

Proof. The hardness proof for the verification (and for the constrained existence) problem is by a reduction from QSAT: for every closed quantified Boolean formula ϕ in conjunctive prenex normal form, we construct a game $\mathcal{G}(\phi)$ with initial state s_1 and safety objectives such that $\mathcal{G}(\phi)$ has a pseudo-Nash equilibrium with payoff $(0, \ldots, 0)$ from s_1 iff the formula is true. Let $\phi = \exists x_1 \forall x_2 \ldots Q_n x_n \; C_1 \wedge C_2 \wedge \cdots \wedge C_m$, where each clause C_j is a disjunction of literals over the variables $x_1, \ldots, x_n$; we identify C_j with the set of literals occurring in it. Then the game $\mathcal{G}(\phi)$ is played by players $0, 1, \ldots, m$. The set of states is $\{s_1, x_1, \neg x_1, \ldots, s_n, x_n, \neg x_n, s_{n+1}\}$, and there are transitions from s_i to x_i and $\neg x_i$, and from x_i and $\neg x_i$ to s_{i+1}; additionally, there is a transition from s_{n+1} back to s_{n+1}. If i is odd, then the state s_i is controlled by player 0; otherwise, the game proceeds nondeterministically from s_i to either x_i or $\neg x_i$ (see Figure 3). Player 0 loses every play of the game, i.e. $\mathcal{L}_0 = \emptyset$, whereas for $j > 0$ player j's objective is to avoid the set of literals occurring in the clause C_j, i.e. $\mathcal{L}_j = (\mathrm{States} \setminus C_j)^\omega$. It is easy to see that ϕ is true iff there is a strategy for player 0 such that all outcomes are losing for all players.

To prove hardness of the existence problem, it suffices to add two states s_0 and s_0' to the game $\mathcal{G}(\phi)$: from s_0, the game proceeds nondeterministically to either s_0' or s_1, and we add

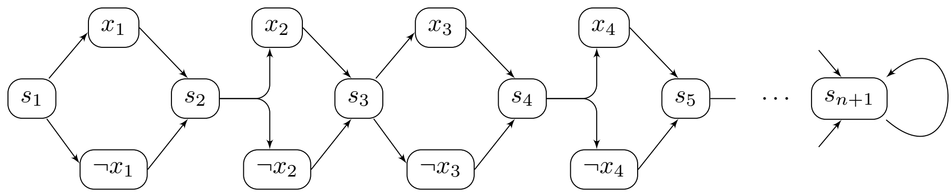

Figure 3 Reducing from QSAT

a transition from s'_0 back to s'_0. Finally, the objective of player 0 is the set $\mathcal{L}_0 = \{s_0, s'_0\}^\omega$. It follows that there is a pseudo-Nash equilibrium from s_0 (with the optimal play leading to s'_0) iff there is a pseudo-Nash equilibrium from s_1 with payoff $(0, \ldots, 0)$. ◀

Note that the hardness proof requires nondeterminism. For deterministic games we can solve the problem by guessing the set of losing players and an (ultimately-periodic) path in the corresponding repellor transition system. We then have to check that all possible deviations fall in some repellor set. The best algorithm we could get for this check runs in time $O(|\text{States}|^2 \cdot |\text{Agt}| \cdot |\text{Tab}|^{\log(\max_i |Q_i|)})$, which is only polynomial when the size of the Büchi automata is bounded.

▶ **Theorem 24.** *The verification, existence, and constrained existence problems are in* NP *for finite deterministic games with objectives defined by* 1*-weak deterministic Büchi automata of bounded size.*

The matching lower bound holds again for the special case of safety objectives since no nondeterministic transitions arise in the construction used for proving Proposition 23 when we reduce from SAT (except for the existence problem, where we require a different construction).

▶ **Proposition 25.** *The verification, existence and constrained existence problems are* NP*-hard for finite deterministic games with safety objectives.*

6 Discussion

In this paper we focused on Büchi objectives. The natural next step is to go to parity objectives, which can encode arbitrary ω-regular objectives. In the turn-based case, the constrained existence problem becomes NP-complete for parity (or even co-Büchi) objectives [19]. In Appendix D, we show that we can get the same upper bound in deterministic concurrent games under the assumption that strategies *can* observe actions.

▶ **Theorem 26.** *The constrained existence problem is in* NP *for finite deterministic concurrent games with parity objectives if we assume that strategies can observe actions.*

In Section 2, we mentioned that making actions unobservable by players is a relevant modelling assumption, but that it makes the computation of equilibria harder. This claim is justified by the following result (proved in Appendix D), which is obtained by a reduction from the strategy problem for *generalised parity games* [7].

▶ **Proposition 27.** *The verification, existence and constrained existence problems are* coNP*-hard for finite deterministic concurrent games with parity objectives. In particular, unless* NP = coNP*, these problems do not belong to* NP.

A natural question is whether the repellor techniques that we develop can be used to handle imperfect information in a more general sense than just state-based vs. action-based strategies. This is one of our directions for future work.

References

1 R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time temporal logic. *J. ACM*, 49(5):672–713, 2002.
2 E. Asarin, O. Maler, A. Pnueli, and J. Sifakis. Controller synthesis for timed automata. In SSC'98, p. 469–474. Elsevier Science, 1998.
3 P. Bouyer, R. Brenguier, and N. Markey. Nash equilibria for reachability objectives in multi-player timed games. In CONCUR'10, LNCS 6269, p. 192–206. Springer, 2010.
4 P. Bouyer, R. Brenguier, and N. Markey. Nash equilibria for reachability objectives in multiplayer timed games. Research Report LSV-10-12, ENS Cachan, France, 2010.
5 K. Chatterjee. Two-player nonzero-sum ω-regular games. In CONCUR'05, LNCS 3653, p. 413–427. Springer, 2005.
6 K. Chatterjee, T. A. Henzinger, and M. Jurdziński. Games with secure equilibria. In LICS'04, p. 160–169. IEEE Comp. Soc. Press, 2004.
7 K. Chatterjee, T. A. Henzinger, and N. Piterman. Generalized parity games. In FoSSaCS'07, LNCS 4423, p. 153–167. Springer, 2007.
8 K. Chatterjee, R. Majumdar, and M. Jurdziński. On Nash equilibria in stochastic games. In CSL'04, LNCS 3210, p. 26–40. Springer, 2004.
9 L. de Alfaro, M. Faella, T. A. Henzinger, R. Majumdar, and M. Stoelinga. The element of surprise in timed games. In CONCUR'03, LNCS 2761, p. 142–156. Springer, 2003.
10 M. Félegyházi, J.-P. Hubaux, and L. Buttyán. Nash equilibria of packet forwarding strategies in wireless ad hoc networks. *IEEE Trans. Mobile Comput.*, 5(5):463–476, 2006.
11 A. M. Fink. Equilibrium in a stochastic n-person game. *J. Science in Hiroshima University*, 28(1):89–93, 1964.
12 D. Fisman, O. Kupferman, and Y. Lustig. Rational synthesis. In TACAS'10, LNCS 6015, p. 190–204. Springer, 2010.
13 T. A. Henzinger. Games in system design and verification. In TARK'05, p. 1–4, 2005.
14 M. Jurdziński, F. Laroussinie, and J. Sproston. Model checking probabilistic timed automata with one or two clocks. In TACAS'07, LNCS 4424, p. 170–184. Springer, 2007.
15 J. F. Nash, Jr. Equilibrium points in n-person games. *Proc. National Academy of Sciences of the USA*, 36(1):48–49, 1950.
16 R. Selten. Spieltheoretische Behandlung eines Oligopolmodells mit Nachfrageträgheit. *Zeitschrift für die gesamte Staatswissenschaft*, 121:301–324 and 667–689, 1965.
17 L. S. Shapley. Stochastic games. *Proc. National Academy of Sciences of the USA*, 39:1095–1100, 1953.
18 W. Thomas. Infinite games and verification (extended abstract of a tutorial). In CAV'02, LNCS 2404, p. 58–64. Springer, 2002.
19 M. Ummels. The complexity of Nash equilibria in infinite multiplayer games. In FoSSaCS'08, LNCS 4962, p. 20–34. Springer, 2008.
20 M. Ummels and D. Wojtczak. The complexity of Nash equilibria in limit-average games. In CONCUR'11, LNCS 6901, p. 482–496. Springer, 2011.
21 M. Ummels and D. Wojtczak. The complexity of Nash equilibria in stochastic multiplayer games. *Logical Methods in Computer Science*, 7(3), 2011.
22 N. Vielle. Two-player stochastic games I–II. *Israel J. of Mathematics*, 119(1):55–126, 2000.

A Appendix for Section 3

A.1 Proof of Lemma 8

▶ **Lemma 8.** *Let $\mathcal{G}$ be a game, $P \subseteq \mathrm{Agt}$, and let ℓ be the length of the longest acyclic path in $\mathcal{G}$. Then $\mathrm{Rep}^n(P) = \mathrm{Rep}^\infty(P)$ for all $n \geq \ell \cdot |P|$.*

Proof. Assume $n \geq 1$ and $s \in \mathrm{Rep}^{n+1}(P) \setminus \mathrm{Rep}^n(P)$. We shall prove that there exists an acyclic path $h \in \mathrm{Hist}(s)$ such that:

- for every $p < |h|$, $h_{=p} \in \mathrm{Rep}^{n+1}(P) \setminus \mathrm{Rep}^n(P)$, and
- either $\mathrm{last}(h) \in \mathrm{Rep}^{n+1}(P') \setminus \mathrm{Rep}^{n+1}(P)$ for some $P' \subsetneq P$ or $\mathrm{last}(h) \in \mathrm{Rep}^n(P') \setminus \mathrm{Rep}^{n-1}(P')$ for some $P' \subseteq P$.

To do so, we define the set S as follows:

$$S = \{s' \in \mathrm{States} \mid \exists h \in \mathrm{Hist}(s) \text{ s.t. } \mathrm{last}(h) = s' \text{ and } s' \in \mathrm{Rep}^{n+1}(P) \setminus \mathrm{Rep}^n(P)\}$$

Assume towards a contradiction that for every $(s', s'') \in S \times \mathrm{States}$ such that $s' \to s''$, for every $P' \subseteq P$, it is the case that $s'' \notin \mathrm{Rep}^n(P') \setminus \mathrm{Rep}^{n-1}(P')$ and $s'' \notin \mathrm{Rep}^{n+1}(P') \setminus \mathrm{Rep}^{n+1}(P)$. Now let $s' \in S$ and $m_{\mathrm{Agt}} \in \mathrm{Secure}^{n+1}(s', P)$. By definition of a secure move, for every $s'' \in \mathrm{States}$, $s'' \in \mathrm{Rep}^{n+1}(P \cap \mathrm{Susp}((s', s''), m_{\mathrm{Agt}}))$ and $s'' \notin \bigcup_{B \in P \cap \mathrm{Susp}((s',s''),m_{\mathrm{Agt}})} \Omega_B \setminus \mathrm{Rep}^n(P \cap \mathrm{Susp}((s', s''), m_{\mathrm{Agt}}))$. Let $P' = P \cap \mathrm{Susp}((s', s''), m_{\mathrm{Agt}})$. We have thus that $s'' \in \mathrm{Rep}^{n+1}(P')$, hence by assumption $s'' \in \mathrm{Rep}^{n+1}(P)$ (because $s'' \notin \mathrm{Rep}^{n+1}(P') \setminus \mathrm{Rep}^{n+1}(P)$). We have also assumed that $s'' \notin \mathrm{Rep}^n(P) \setminus \mathrm{Rep}^{n-1}(P)$, hence gathering everything, we get that $s'' \in \mathrm{Rep}^{n+1}(P) \setminus \mathrm{Rep}^n(P)$ or $s'' \in \mathrm{Rep}^{n-1}(P)$. Hence $s'' \in S \cup \mathrm{Rep}^{n-1}(P)$.

Now from $s'' \notin \bigcup_{B \in P'} \Omega_B \setminus \mathrm{Rep}^n(P')$, we also get that either $s'' \notin \bigcup_{B \in P'} \Omega_B$ or $s'' \in \mathrm{Rep}^n(P')$. The second case, together with the assumption that $s'' \notin \mathrm{Rep}^n(P') \setminus \mathrm{Rep}^{n-1}(P')$, implies that $s'' \in \mathrm{Rep}^{n-1}(P')$. Therefore if $s'' \in \Omega_B$ for some $B \in P'$ then $s'' \in \mathrm{Rep}^{n-1}(P')$.

This proves that set S satisfies the fixpoint condition defining $\mathrm{Rep}^n(P)$. Hence $S \subseteq \mathrm{Rep}^n(P)$, which is a contradiction since $s \in S \cap (\mathrm{Rep}^{n+1}(P) \setminus \mathrm{Rep}^n(P))$.

We conclude from this that there exists some $(s', s'') \in S \times \mathrm{States}$ with $s' \to s''$ and there exists some $P' \subseteq P$ such that either $s'' \in \mathrm{Rep}^n(P') \setminus \mathrm{Rep}^{n-1}(P')$ or $s'' \in \mathrm{Rep}^{n+1}(P') \setminus \mathrm{Rep}^{n+1}(P)$. Note that the second case can only happen if $P' \subsetneq P$. Hence there exists some history $h \in \mathrm{Hist}(s)$ (and we can assume it is acyclic) such that for all $p < |h|$, $h_{=p} \in \mathrm{Rep}^{n+1}(P) \setminus \mathrm{Rep}^n(P)$, and either $\mathrm{last}(h) \in \mathrm{Rep}^n(P') \setminus \mathrm{Rep}^{n-1}(P')$ for some $P' \subsetneq P$ or $\mathrm{last}(h) \in \mathrm{Rep}^{n+1}(P') \subseteq \mathrm{Rep}^{n+1}(P)$ for some $P' \subsetneq P$.

We will now prove the claimed bounds. Assume towards a contradiction that $s \in \mathrm{Rep}^{n+1}(P) \setminus \mathrm{Rep}^n(P)$ with $n \geq \ell \cdot |P|$ (resp. $n \geq \ell$ in case the game is finite), and define the pair $(n_1, P_1) = (n, P)$. From the previous construction, we are able to construct a sequence of pairs $((n_i, P_i))_{1 \leq i \leq k}$ such that for every $1 \leq i < k$:

- either $n_{i+1} = n_i - 1$ and $P_{i+1} \subseteq P_i$,
- or $n_{i+1} = n_i$ and $P_{i+1} \subsetneq P_i$,
- $n_k = 1$,

and a finite path $h = h_1 \cdot h_2 \dots h_k \in \mathrm{Hist}(s)$ such that:

- for all $1 \leq i < k$, h_i is acyclic and all states along h_i belong to $\mathrm{Rep}^{n_i}(P_i) \setminus \mathrm{Rep}^{n_{i-1}}(P_{i-1})$,
- h_k has a single state which belongs to $\mathrm{Rep}^{n_k}(P_k)$.

For every $1 \le i \le k$ we write S_i for the set of states appearing along h_i. By the acyclicity of the h_i's, $|S_i| = |h_i|$. We fix some $P' \subseteq P$, and we write $J(P')$ for the set of indices j such that $P_j = P'$. It is a bounded interval of $\mathbb{N}_+$ (which may be empty). A state s cannot belong to two different $\mathrm{Rep}^{m+1}(P') \setminus \mathrm{Rep}^{m}(P')$ (with the same P') since the sequence $(\mathrm{Rep}^m(P'))_m$ is non-decreasing. Hence $\sum_{j\in J(P')} |S_j| \le \ell$. We finally notice that $|h| = \sum_{i=1}^k |h_i| = \sum_{i=1}^k |S_i| = \sum_{P'\subseteq P} \sum_{j\in J(P')} |S_j| \le |P| \cdot \ell$ since the number of P''s for which $J(P')$ is not empty is bounded by $|P|$. Furthermore there is at least one state along h for each index $1 \le m \le n+1$, which implies that $|h| \ge n+1$. There is thus a contradiction with the assumption that $n \ge \ell \cdot |P|$. Hence, $\mathrm{Rep}^\infty(P) = \mathrm{Rep}^{n+1}(P) = \mathrm{Rep}^n(P)$. ◀ ◀

A.2 Proof of Proposition 9

▶ **Proposition 9.** *Let $\mathcal{G}$ be a finite game, $P \subseteq \mathrm{Agt}$, and let $\rho \in \mathrm{Play}(s)$ be a play that visits $\bigcup_{B\in P} \Omega_B$ only finitely often. Then ρ is a path in $\mathcal{S}^\infty(P)$ if and only if there exists $\sigma_{\mathrm{Agt}} \in \mathrm{Prof}$ such that $\rho \in \mathrm{Out}(s, \sigma_{\mathrm{Agt}})$ and ρ' does not visit Ω_B infinitely often for all plays ρ' that can arise when some player $B \in P$ changes her strategy, i.e. $\rho' \in \mathrm{Out}(s, \sigma_{\mathrm{Agt}}[B \mapsto \sigma'])$ for some $B \in P$ and some $\sigma' \in \mathrm{Strat}^B$.*

The proof of this proposition relies on several preliminary lemmas. We first extend the notion of suspect players from edges to finite paths $\pi = (s_k)_{k\le|\pi|}$ by setting

$$\mathrm{Susp}(\pi, \sigma_{\mathrm{Agt}}) = \bigcap_{k<|\pi|} \mathrm{Susp}((s_k, s_{k+1}), \langle\sigma_A(\pi_{\le k})\rangle_{A\in\mathrm{Agt}}) .$$

▶ **Lemma 28.** *Let $P \subseteq \mathrm{Agt}$, n be a positive integer and $s \in \mathrm{States}$. Let $\sigma_{\mathrm{Agt}} \in \mathrm{Prof}$, and assume that for all $h \in \mathrm{Hist}(s)$ s.t. $|h| > 0$, $\mathrm{last}(h)$ does not belong to*

$$\bigcup_{B\in P\cap\mathrm{Susp}(h,\sigma_{\mathrm{Agt}})} \Omega_B \setminus \mathrm{Rep}^{n-1}(P \cap \mathrm{Susp}(h, \sigma_{\mathrm{Agt}})).$$

Then for every $h \in \mathrm{Hist}(s)$:

- $\mathrm{last}(h)$ *belongs to* $\mathrm{Rep}^n(P \cap \mathrm{Susp}(h, \sigma_{\mathrm{Agt}}))$*, and*
- $\sigma_{\mathrm{Agt}}(h) \in \mathrm{Secure}^n(\mathrm{last}(h), P \cap \mathrm{Susp}(h, \sigma_{\mathrm{Agt}}))$.

In particular, any $\rho \in \mathrm{Out}(s, \sigma_{\mathrm{Agt}})$ is a path in $\mathcal{S}^n(P)$. Also, $s \in \mathrm{Rep}^n(P)$.

Proof. We fix the profile σ_{Agt}, and we aim at proving that for every $h \in \mathrm{Hist}(s)$, $\mathrm{last}(h) \in \mathrm{Rep}^n(P \cap \mathrm{Susp}(h, \sigma_{\mathrm{Agt}}))$ and $\sigma_{\mathrm{Agt}}(h) \in \mathrm{Secure}^n(P \cap \mathrm{Susp}(h, \sigma_{\mathrm{Agt}}))$. By the definition of a secure move, this last property will be a direct consequence of the former property.

To this aim, we define for every $P \subseteq \mathrm{Agt}$ the set

$$S_P = \{s' \in \mathrm{States} \mid \exists h \in \mathrm{Hist}(s).\ \mathrm{last}(h) = s' \text{ and } P \subseteq \mathrm{Susp}(h, \sigma_{\mathrm{Agt}})\}$$

and we prove by induction on P that $S_P \subseteq \mathrm{Rep}^n(P)$. The case $P = \emptyset$ is straightforward since $\mathrm{Rep}^n(P) = \mathrm{States}$. If $P \ne \emptyset$ we assume that the result holds true for all $P' \subsetneq P$. Pick now $s' \in S_P$ with h as a witness, and define $m_{\mathrm{Agt}} = \sigma_{\mathrm{Agt}}(h)$. Take some $s'' \in \mathrm{States}$. By hypothesis, we have that $s'' \notin \bigcup_{B\in P\cap\mathrm{Susp}(h.s'',\sigma_{\mathrm{Agt}})} \Omega_B \setminus \mathrm{Rep}^{n-1}(P \cap \mathrm{Susp}(h.s'', \sigma_{\mathrm{Agt}}))$. As $P \subseteq \mathrm{Susp}(h, \sigma_{\mathrm{Agt}})$, we get that $P \cap \mathrm{Susp}(h.s'', \sigma_{\mathrm{Agt}}) = P \cap \mathrm{Susp}((s', s''), m_{\mathrm{Agt}})$. In particular, $s'' \notin \bigcup_{B\in P\cap\mathrm{Susp}((s',s''),m_{\mathrm{Agt}})} \Omega_B \setminus \mathrm{Rep}^{n-1}(P \cap \mathrm{Susp}((s', s''), m_{\mathrm{Agt}}))$. We distinguish now between two cases:

- If $P \subseteq \mathrm{Susp}((s', s''), m_{\mathrm{Agt}})$, then we get that $s'' \in S_P$ (by definition of the set S_P).

- If $P \cap \mathrm{Susp}((s', s''), m_{\mathrm{Agt}}) \subsetneq P$, we define $P' = P \cap \mathrm{Susp}((s', s''), m_{\mathrm{Agt}})$, and we get that $s'' \in S_{P'}$. Hence, by induction hypothesis, we get that $s'' \in \mathrm{Rep}^n(P')$.

As $\mathrm{Rep}^n(P)$ is the greatest fixpoint which satisfies the above conditions, we get that $S_P \subseteq \mathrm{Rep}^n(P)$, which concludes the proof of the lemma. ◀ ◀

▶ **Lemma 29.** *Let $P \subseteq \mathrm{Agt}$ and n be a positive integer. Assume that ρ is an infinite path in $\mathcal{S}^n(P)$. Then there exists $\sigma_{\mathrm{Agt}} \in \mathrm{Prof}$ such that $\rho \in \mathrm{Out}(s, \sigma_{\mathrm{Agt}})$ and for all $h \in \mathrm{Hist}(s)$ such that $|h| > 0$, $\mathrm{last}(h)$ does not belong to*

$$\bigcup_{B \in P \cap \mathrm{Susp}(h, \sigma_{\mathrm{Agt}})} \Omega_B \setminus \mathrm{Rep}^{n-1}(P \cap \mathrm{Susp}(h, \sigma_{\mathrm{Agt}})).$$

Proof. The proof is quite similar to the proof of Proposition 10 in [A1].

We assume that $\rho = (s_p)_{p \geq 0}$ is a play in $\mathcal{S}^n(P)$ from $s \in \mathrm{Rep}^n(P)$, and we define the following two (partial) functions:

- $c\colon \mathrm{States} \times 2^{\mathrm{Agt}} \to \mathrm{Act}^{\mathrm{Agt}}$: for all (s, Q) such that $s \in \mathrm{Rep}^n(Q)$, we let $c(s, Q) = m_{\mathrm{Agt}}$ for some $m_{\mathrm{Agt}} \in \mathrm{Secure}^n(s, Q)$; for all (s, Q) such that $s \notin \mathrm{Rep}^n(Q)$, we let $c(s, Q) = m_{\mathrm{Agt}}$ for some $m_{\mathrm{Agt}} \in \mathrm{Act}^{\mathrm{Agt}}$.
- $d_\rho\colon \mathbb{N} \to \mathrm{Act}^{\mathrm{Agt}}$: for all $p \in \mathbb{N}$, we let $d_\rho(p) = m_{\mathrm{Agt}}$ for some $m_{\mathrm{Agt}} \in \mathrm{Secure}^n(s_p, P)$ such that $(s_p, s_{p+1}) \in \mathrm{Tab}(s_p, m_{\mathrm{Agt}})$. This is well-defined because ρ is a play in $\mathcal{S}^n(P)$.

The strategy profile σ_{Agt} is then defined as follows:

- on prefixes of ρ, we let $\sigma_A(\rho_{\leq p}) = (d_\rho(p))(A)$ for all $p \in \mathbb{N}$ and all $A \in \mathrm{Agt}$;
- for any $h = (s'_p)_{p \leq |h|}$ that is not a prefix of ρ, for every $A \in \mathrm{Agt}$ we let

$$\sigma_A(h) = c(s'_{|h|}, P \cap \mathrm{Susp}(h, \sigma_{\mathrm{Agt}}))(A).$$

By construction, we have that $\rho \in \mathrm{Out}(s, \sigma_{\mathrm{Agt}})$.

Take now $\rho' = (s'_p)_{p \geq 0} \in \mathrm{Play}(s)$. We show by induction on p that $s'_p \in \mathrm{Rep}^n(P \cap \mathrm{Susp}(\rho'_{\leq p}), \sigma_{\mathrm{Agt}})$, and that for any $B \in P \cap \mathrm{Susp}(\rho'_{\leq p+1}, \sigma_{\mathrm{Agt}})$, s'_{p+1} does not belong to $\Omega_B \setminus \mathrm{Rep}^{n-1}(P \cap \mathrm{Susp}(\rho'_{\leq p+1}, \sigma_{\mathrm{Agt}}))$. This clearly holds when $p = 0$ (as $s = s'_0 \in \mathrm{Rep}^n(P)$). Let us assume that it holds for some $p \geq 0$. Let $m_A = \sigma_A(\rho'_{\leq p})$ for all $A \in \mathrm{Agt}$.

- If $\rho'_{\leq p}$ is not a prefix of ρ: the strategies σ_A are given by c, and as $s'_p \in \mathrm{Rep}^n(P \cap \mathrm{Susp}(\rho'_{\leq p}, \sigma_{\mathrm{Agt}}))$, we get $m_{\mathrm{Agt}} \in \mathrm{Secure}^n(s'_p, P \cap \mathrm{Susp}(\rho'_{\leq p}, \sigma_{\mathrm{Agt}}))$.
- If $\rho'_{\leq p}$ is a prefix of ρ: the strategies σ_A are given by d_ρ, and it is the case that m_{Agt} is in $\mathrm{Secure}^n(s'_p, P) = \mathrm{Secure}^n(s'_p, P \cap \mathrm{Susp}(\rho'_{\leq p}, \sigma_{\mathrm{Agt}}))$ (because $\rho'_{\leq p}$ is an outcome of σ_{Agt}, hence $\mathrm{Susp}(\rho'_{\leq p}, \sigma_{\mathrm{Agt}}) = \mathrm{Agt}$).

Thus, in both cases, we get that

$$s'_{p+1} \in \mathrm{Rep}^n(P \cap \mathrm{Susp}(\rho'_{\leq p}, \sigma_{\mathrm{Agt}}) \cap \mathrm{Susp}((s'_p, s'_{p+1}), m_{\mathrm{Agt}})) = \mathrm{Rep}^n(P \cap \mathrm{Susp}(\rho'_{\leq p+1}, \sigma_{\mathrm{Agt}}))$$

Also it is the case that $s'_{p+1} \notin \Omega_B \setminus \mathrm{Rep}^{n-1}(P \cap \mathrm{Susp}(\rho'_{\leq p}, \sigma_{\mathrm{Agt}}) \cap \mathrm{Susp}((s'_p, s'_{p+1}), m_{\mathrm{Agt}})) = \Omega_B \setminus \mathrm{Rep}^{n-1}(P \cap \mathrm{Susp}(\rho'_{\leq p+1}, \sigma_{\mathrm{Agt}}))$, for any $B \in P$. This proves the inductive step, and concludes the proof of the lemma. ◀ ◀

▶ **Corollary 30.** *Let $P \subseteq \mathrm{Agt}$, n be a positive integer and $s \in \mathrm{States}$. Then $s \in \mathrm{Rep}^n(P)$ if and only if there exists $\sigma_{\mathrm{Agt}} \in \mathrm{Prof}$ such that for all $h \in \mathrm{Hist}(s)$ such that $|h| > 0$, $\mathrm{last}(h)$ does not belong to*

$$\bigcup_{B \in P \cap \mathrm{Susp}(h, \sigma_{\mathrm{Agt}})} \Omega_B \setminus \mathrm{Rep}^{n-1}(P \cap \mathrm{Susp}(h, \sigma_{\mathrm{Agt}})).$$

Proof. ($\Rightarrow$) If $s \in \mathrm{Rep}^n(P)$, then there exists some infinite play ρ in $\mathcal{S}^n(P)$. We can thus apply Lemma 29 and we get the expected result.
($\Leftarrow$) We can apply Lemma 28 to the strategy profile and we get in particular that $s \in \mathrm{Rep}^n(P)$. ◀ ◀

▶ **Lemma 31.** *Let $P \subseteq \mathrm{Agt}$, $s \in \mathrm{States}$, and $\sigma_{\mathrm{Agt}} \in \mathrm{Prof}$ be such that for all $B \in P$, for all $\sigma' \in \mathrm{Strat}^B$ it holds that:*

$$\forall \rho \in \mathrm{Out}(s, \sigma_{\mathrm{Agt}}[B \mapsto \sigma']).\ \rho \textit{ does not visit } \Omega_B \textit{ infinitely often}$$

Then for all $h \in \mathrm{Hist}(s)$ s.t. $|h| > 0$, $\mathrm{last}(h)$ does not belong to

$$\bigcup_{B \in P \cap \mathrm{Susp}(h, \sigma_{\mathrm{Agt}})} \Omega_B \setminus \mathrm{Rep}^\infty(P \cap \mathrm{Susp}(h, \sigma_{\mathrm{Agt}})).$$

Proof. We do the proof by induction on P. The property is obviously true if $P = \emptyset$ and we assume that it holds for all $P' \subsetneq P$. Consider a strategy profile σ_{Agt} as given in the statement of the proposition, and define the set H as follows:

$$H = \{h \in \mathrm{Hist}(s) \mid |h| > 0 \text{ and } \mathrm{last}(h) \in \bigcup_{B \in P \cap \mathrm{Susp}(h, \sigma_{\mathrm{Agt}})} \Omega_B \setminus \mathrm{Rep}^\infty(P \cap \mathrm{Susp}(h, \sigma_{\mathrm{Agt}}))\}$$

Assume towards a contradiction that $H \neq \emptyset$, and write for every $h \in H$ the set $P_h = P \cap \mathrm{Susp}(h, \sigma_{\mathrm{Agt}})$. We distinguish between two cases:

- If there exists $h \in H$ such that $P_h \subsetneq P$, then by induction hypothesis applied to P_h, s and σ_{Agt}, we get that $\mathrm{last}(h)$ should not belong to $\bigcup_{B \in P_h} \Omega_B \setminus \mathrm{Rep}^\infty(P_h \cap \mathrm{Susp}(h, \sigma_{\mathrm{Agt}}))$. This is a contradiction, as $P_h \cap \mathrm{Susp}(h, \sigma_{\mathrm{Agt}}) = P_h$.
- We assume that for every $h \in H$, $P_h = P$, that is $P \subseteq \mathrm{Susp}(h, \sigma_{\mathrm{Agt}})$. We have that for every $h \in H$, $\mathrm{last}(h) \notin \mathrm{Rep}^\infty(P)$ and there exists some $B_h \in P$ such that $\mathrm{last}(h) \in \Omega_{B_h}$. We now build finite sequences of histories $(h_i)_{1 \le i \le n}$ (for $n \ge 1$) such that for every $1 \le i \le n$, $h_i \in H$ and for every $1 \le i < n$, h_i is a strict prefix of h_{i+1}. We do this by induction on n. The case $n = 1$ is straightforward by assumption ($H \neq \emptyset$), and we now assume that we have constructed such a sequence for some $n \ge 1$. We define the strategy profile σ'_{Agt} by $\sigma'_{\mathrm{Agt}}(h) = \sigma_{\mathrm{Agt}}(h_n \cdot h_{>1})$ if $\mathrm{first}(h) = \mathrm{last}(h_n)$, and $\sigma'_{\mathrm{Agt}}(h)$ is anything otherwise. We now choose m such that for every $P' \subseteq \mathrm{Agt}$, $\mathrm{Rep}^{m-1}(P') = \mathrm{Rep}^m(P') = \mathrm{Rep}^\infty(P')$. We apply Corollary 30 to P, $\mathrm{last}(h_n)$ (which is not in $\mathrm{Rep}^m(P)$ by assumption) and σ'_{Agt}. There exists some $h \in \mathrm{Hist}(\mathrm{last}(h_n))$ with $|h| > 0$ such that $\mathrm{last}(h) \in \bigcup_{B \in P \cap \mathrm{Susp}(h, \sigma'_{\mathrm{Agt}})} \Omega_B \setminus \mathrm{Rep}^\infty(P \cap \mathrm{Susp}(h, \sigma'_{\mathrm{Agt}}))$. We then set $h_{n+1} = h_n \cdot h_{>1}$, and we get easily that $h_{n+1} \in H$, and it satisfies the condition for proving the induction step.
  Recall that we for every $n \ge 1$, $P_{h_n} = P$, hence the limit $\rho = \lim_{n \to \infty} h_n$ visits $\bigcup_{B \in P} \Omega_B \setminus \mathrm{Rep}^\infty(P)$ infinitely often. Hence we are able to choose $B \in P$ such that ρ visits Ω_B infinitely often, and for that precise B (which is in $P = P \cap \mathrm{Susp}(\rho, \sigma_{\mathrm{Agt}})$), we can choose a strategy $\sigma' \in \mathrm{Strat}^B$ such that $\rho \in \mathrm{Out}(s, \sigma_{\mathrm{Agt}}[B \mapsto \sigma'])$. This contradicts the assumption, and concludes the proof of the lemma. ◀

◀

of Proposition 9. ($\Rightarrow$) Assume ρ is a path in $\mathcal{S}^\infty(P)$. As $\mathcal{G}$ is finite, there is an integer n such that ρ is a path in $\mathcal{S}^n(P)$. We construct a strategy profile $\sigma(n,P,\rho)_{\mathrm{Agt}}$ in the following way.

We first fix a strategy profile $\sigma^{n,P,\rho}_{\mathrm{Agt}}$ which satisfies the conditions of Lemma 29 for n, P and ρ. We fix some history h:

1. If h is a prefix of ρ or if h does not visit $\bigcup_{B\in P}\Omega_B$, we set $\sigma(n,P,\rho)_{\mathrm{Agt}}(h) = \sigma^{n,P,\rho}_{\mathrm{Agt}}(h)$;
2. Otherwise we assume the first time h visits $\bigcup_{B\in P}\Omega_B$ is at p-th state $h_{=p}$, and we define $P' = P \cap \mathrm{Susp}(h_{\le p}, \sigma^{n,P,\rho}_{\mathrm{Agt}})$. We distinguish between several cases:
   a. if $P' = \emptyset$, then we set $\sigma(n,P,\rho)_{\mathrm{Agt}}(h)$ to be $\sigma^{n,P,\rho}_{\mathrm{Agt}}(h)$;
   b. if $h_{=p} \in \mathrm{Rep}^{n-1}(P')$,[1] then we choose some infinite path ρ' which starts from $h_{=p}$ and is in $\mathcal{S}^{n-1}(P')$. We then set $\sigma(n,P,\rho)_{\mathrm{Agt}}(h)$ to $\sigma(n-1,P',\rho')_{\mathrm{Agt}}(h_{\ge p})$.
   c. otherwise, it is the case that $\mathrm{last}(h) \in \bigcup_{B\in P\setminus P'}\Omega_B$. Applying Lemma 28 we get that $\mathrm{last}(h) \in \mathrm{Rep}^n(P')$ with $P' \subsetneq P$. We then fix any infinite path ρ' in $\mathcal{S}^n(P')$ from $h_{=p}$ and we set $\sigma(n,P,\rho)_{\mathrm{Agt}}(h)$ to $\sigma(n,P',\rho')_{\mathrm{Agt}}(h_{\ge p})$.

This is well-defined since case 2.(b) cannot happen with $n=1$ since $\mathrm{Rep}^0(P') = \emptyset$ for every $P' \subseteq \mathrm{Agt}$. By construction, and since $\rho \in \mathrm{Out}(s, \sigma^{n,P,\rho}_{\mathrm{Agt}})$, we get that $\rho \in \mathrm{Out}(s, \sigma(n,P,\rho)_{\mathrm{Agt}})$. And by hypothesis, for every $B \in P$, ρ does not visit Ω_B infinitely often. We write b for the number of visits of $\bigcup_{B\in P}\Omega_B$ along ρ.

We prove now by induction over n and P that $\sigma(n,P,\rho)_{\mathrm{Agt}}$ is such that for all $B \in P$, all $\sigma' \in \mathrm{Strat}^B$, and for all $\rho' \in \mathrm{Out}(s, \sigma(n,P,\rho)_{\mathrm{Agt}}[B \mapsto \sigma'])$, if $\rho'_{\le p}$ is the longest common prefix of ρ' with ρ, ρ'_p does not visit Ω_B more than $n + |P| - 1$ times. Hence if this is true, globally ρ' will not visit Ω_B more than $n + |P| - 1 + b$ times.

Note first that if $P = \emptyset$, then $\sigma(n,\emptyset,\rho)_{\mathrm{Agt}}$ is defined everywhere as $\sigma^{n,\emptyset,\rho}_{\mathrm{Agt}}$, and the property obviously holds (no constraints on visits to Ω_B's).

Assume now that $n = 1$. We prove the property by induction on P, the case where $P = \emptyset$ has already been handled. Thus assume that it holds for every $P' \subsetneq P$. Fix some $B \in P$ and take $\rho' \in \mathrm{Out}(s, \sigma(1,P,\rho)_{\mathrm{Agt}}[B \mapsto \sigma'])$ for some $\sigma' \in \mathrm{Strat}^B$. Define integer p such that $\rho'_{\le p}$ is the longest common prefix of ρ' with ρ.

- If $\rho'_{>p}$ does not visit $\bigcup_{B\in P}\Omega_B$, then we are happy.
- Otherwise, assume that $\rho'_{=p'}$ is the first visit to $\bigcup_{B\in P}\Omega_B$ in $\rho'_{>p}$. When constructing the strategy, case 2.(a) cannot happen since ρ' is an outcome of $\sigma(1,P,\rho)_{\mathrm{Agt}}[B \mapsto \sigma']$, and case 2.($b$) cannot happen either since $\mathrm{Rep}^0(P')$ is empty for every P'. Hence by construction, $\rho'_{\ge p'} \in \mathrm{Out}(\rho'_{=p'}, \sigma(1,P',\rho'))$ with $P' \subsetneq P$. By induction hypothesis, the number of visits to $\bigcup_{B\in P}\Omega_B$ for $\rho'_{>p'}$ is bounded by $1 + |P'| - 1 = |P'|$. The number of such visits for $\rho'_{>p}$ is thus one more, that is, is bounded by $|P'| + 1 \le |P|$.

This concludes the proof of the case $n = 1$.

We assume $n > 1$ and $P \ne \emptyset$. And we assume that the property holds either if $n' < n$ or if $P' \subsetneq P$. Fix some $B \in P$ and take $\rho' \in \mathrm{Out}(s, \sigma(n,P,\rho)_{\mathrm{Agt}}[B \mapsto \sigma'])$ for some $\sigma' \in \mathrm{Strat}^B$. Define integer p such that $\rho'_{\le p}$ is the longest common prefix of ρ' with ρ.

- If $\rho'_{>p}$ does not visit $\bigcup_{B\in P}\Omega_B$, then we are happy.
- Otherwise, assume that $\rho'_{=p'}$ is the first visit to $\bigcup_{B\in P}\Omega_B$ in $\rho'_{>p}$. When constructing the strategy, case 2.(a) cannot happen since ρ' is an outcome of $\sigma(n,P,\rho)_{\mathrm{Agt}}[B \mapsto \sigma']$.

[1] This is for instance the case if $h_{\le p} \in \mathrm{Out}^{\mathrm{f}}(\sigma^{n,P,\rho}_{\mathrm{Agt}})$ since in that case $P' = P$ and by Lemma 29, $h_{=p}$ does not belong to $\bigcup_{B\in P}\Omega_B \setminus \mathrm{Rep}^{n-1}(P)$.

- Case 2.(b): $\rho'_{\geq p'} \in \mathrm{Out}(\rho'_{=p'}, \sigma(n-1, P', \rho'))$ for some $P' \subseteq P$. By induction hypothesis, we get that the number of visits to $\bigcup_{B \in P} \Omega_B$ of $\rho'_{>p'}$ is bounded by $(n-1) + |P'| - 1$, hence the number of such visits along $\rho'_{>p}$ is one more, that is is bounded by $1 + (n-1) + |P'| - 1 \leq n + |P| - 1$.
- Case 2.(c): $\rho'_{\geq p'} \in \mathrm{Out}(\rho'_{=p'}, \sigma(n, P', \rho'))$ with $P' \subsetneq P$, and a similar reasoning gives the expected result.

This concludes the proof of the induction step.

($\Leftarrow$) This is a direct application of Lemma 31, Lemma 29 and Lemma 28, where n is large enough so that $\mathrm{Rep}^{n-1}(P) = \mathrm{Rep}^{\infty}(P)$ for every $P \subseteq \mathrm{Agt}$. ◀ ◀

A.3 Proof of the lower bound in Theorem 11

This part is inspired by the proof of [A1, Proposition 13(1)].

The lower bound for the verification problem is straightforward since the circuit value problem can be encoded easily into a deterministic turn-based game: a circuit (which we assume w.l.o.g. has only and- and or-gates) is transformed into a two-player turn-based game, where one player controls the and-gates and the other player controls the or-gates. We add self-loops on the leaves. Positive leaves of the circuit are the (Büchi) objective of the or-player, and negative leaves are the (Büchi) objective of the and-player. Then obviously, the circuit evaluates to true iff the or-player has a winning strategy for satisfying his Büchi condition, which in turn is equivalent to the fact that there is an equilibrium with payoff 1 for Player or and payoff 0 for Player and.

Such a reduction is not possible for the existence problem since in turn-based deterministic games, there is always a Nash equilibrium [A3]. However a slight twist of the construction will be sufficient to get a PTIME lower bound for the existence problem in the context of either turn-based nondeterministic games or concurrent deterministic games. The twist consists in adding a small module from the and-leaves where there is no equilibrium, instead of the self-loop mentioned in the previous reduction. In this case, if there is an equilibrium then it has payoff 1 for Player or and payoff 0 for Player and. Such a module can be easily built, for instance the matching-pennies game in the case of concurrent games.

A.4 Computation of $\mathcal{S}^{\infty}(P)$ in polynomial time

In this section we prove the following lemma:

▶ **Lemma 32.** *Let $P \subseteq \mathrm{Agt}$. Then the transition system $\mathcal{S}^{\infty}(P)$ can be computed in polynomial time.*

Proof. This proof is very similar to the proof of [A1, Proposition 13(2.)].

We have that $\mathcal{S}^n(P) \subseteq \mathcal{S}^{n+1}(P)$ and that $\mathcal{S}^{\infty}(P) = \mathcal{S}^{\ell \cdot |P|}(P)$ (Lemma 8) where ℓ is the length of the longest acyclic path in $\mathcal{G}$.

The procedure consists in filling a table $R[s, n, P]$ where s ranges over States, $n \in \{0, 1, \ldots, \ell \cdot |P|\}$, and P ranges over 2^{Agt}, with the intended meaning that $R[s, n, P] = 1$ if $s \in \mathrm{Rep}^n(P)$, and $R[s, n, P] = 0$ otherwise. This is achieved inductively as follows: $R[s, 0, P] = 0$ for every $P \subseteq \mathrm{Agt}$ and $R[s, n, \emptyset] = 1$ for every $n \in \{1, \ldots, \ell \cdot |P|\}$.

Now, given $P \neq \emptyset$ and $n \in \{1, \ldots, \ell \cdot |P|\}$, and assuming that $R[s, n', P']$ has been computed for any (n', P') such that either $n' < n$ or $P' \subsetneq P$, we compute $R[s, n, P]$ using a standard procedure for computing greatest fixpoints, following the definition of the n-th repellor (see Section 3.1): start with letting $R[s, n, P] = 1$ for all $s \in$ States, and iteratively set $R[s, n, P]$ to 0 for states that do not satisfy the condition on the existence of a secure move. This is achieved by enumerating the set of moves $\langle m_A \rangle_{A \in \mathrm{Agt}}$ (by scanning the transition table), and for each state computing the set of suspects and checking that the state belongs to the corresponding repellor set. This procedure computes the repellor set $\mathrm{Rep}^n(P)$ in at most $|\mathrm{States}|$ steps (assuming the 'predecessors' in the table has been filled), each step being quadratic in the size of the transition table and in the number of states. Moreover, during this computation, we can also get the set of secure moves from each state, thus building the transition system $\mathcal{S}^n(P)$.

However, since there may be exponentially many subsets P' of P, it might take exponential time to compute $\mathrm{Rep}^n(P)$. We make it run in polynomial time, by proving that we need only polynomially many subsets of P during the computation of $\mathrm{Rep}^n(P)$. Hence we don't need to fill the whole table R to get the value of $R[s, n, P]$ and to construct $\mathcal{S}^n(P)$. We begin with defining this set, inductively. This uses the following sets, for each $s \in \mathrm{States}$:

$$\mathfrak{S}_s = \{\mathrm{Susp}((s, s'), m_{\mathrm{Agt}}) \mid (s, s') \notin \mathrm{Tab}(s, m_{\mathrm{Agt}})\}$$

First remark that those sets are not too big: let $U_s = \bigcup_{P' \in \mathfrak{S}_s} P'$; being a suspect for some move m_{Agt}, any player B in U_s must have at least a second action m', besides m_B, for which $(s, s') \in \mathrm{Tab}(s, m_{\mathrm{Agt}}[B \mapsto m'])$. Thus the transition table from state s has size at least $2^{|U_s|}$. On the other hand, $\mathfrak{S}_s$ contains subsets of players from U_s, and thus has size at most $2^{|U_s|}$, hence $|\mathfrak{S}_s| \leq |\mathrm{Tab}(s)|$. For each $s \in \mathrm{States}$, we let n_s be the size of the largest set of suspects in $\mathfrak{S}_s$. Using the same argument as above, each player in any set of $\mathfrak{S}_s$ has at least two actions from s, so that $2^{n_s} \leq |\mathrm{Tab}(s)|$.

Now, the computation of $R[s, n, P]$ for all s involves the following sets of players:

- $\mathfrak{P}_0 = \{P\}$, the initial set,
- for each $P_i \in \mathfrak{P}_i$, computing $R[s', n', P_i]$ (possibly for all $s' \in \mathrm{States}$ and all $n' \leq n$) involves $\{P_i \cap S \mid S \in \mathfrak{S}_{s'}\} \setminus \{P_i\}$. We let

$$\mathfrak{P}_{i+1} = \bigcup_{P_i \in \mathfrak{P}_i, s' \in \mathrm{States}} \{P_i \cap S \mid S \in \mathfrak{S}_{s'}\} \setminus \{P_i\}.$$

  Notice that any set in $\mathfrak{P}_i$ has size at most $|P| - i$, so that $\mathfrak{P}_{|P|+1}$ is empty (and $\mathfrak{P}_{|P|}$ contains at most the empty set).

In order to compute $R[s, n, P]$, it is sufficient to compute $R[s', n', P']$ for all $P' \in \bigcup_{i \leq |P|} \mathfrak{P}_i$ and $n' \leq n$. We now show that this union contains polynomially many sets. By definition of $\mathfrak{P}_i$, any $P' \in \mathfrak{P}_i$ is an intersection of P and of one or several sets in some of the $\mathfrak{S}_{s''}$:

$$\exists (J_s)_{s \in \mathrm{States}}.\ (\forall s \in \mathrm{States}.\ J_s \subseteq \mathfrak{S}_s) \text{ and } P' = P \cap \bigcap_{\substack{s \in \mathrm{States} \\ J_s \neq \emptyset}} \bigcap_{P'' \in J_s} P''$$

For $P' \neq P$, let s_0 be the[2] state for which $n_{s_0} = \min\{n_s \mid J_s \neq \emptyset\}$. Then $|P'| \leq n_{s_0}$, and P' is a subset of some set in $\mathfrak{S}_{s_0}$. Hence, for this s_0, there are at most $2^{n_{s_0}} \times |\mathfrak{S}_{s_0}|$ different possible P' occurring in some $\mathfrak{P}_i$. As a consequence, the total number of sets in $\bigcup_{i \leq |P|} \mathfrak{P}_i$ is at most

$$\sum_{s \in \mathrm{States}} 2^{n_s} \times |\mathfrak{S}_s| \leq |\mathrm{Tab}|^2.$$

[2] In order to have s_0 uniquely defined, we assume an ordering of the states in States, and pick the smallest set in case several would match.

In the end, this shows that only cells $R[\cdot, n', P']$ with $P' \in \bigcup_{i \le |P|} \mathfrak{P}_i$ and $n' \le n$ (with $(n', P') \neq (n, P)$) need to be filled before we are able to fill $R[\cdot, n, P]$, and they can be filled "on-demand". Remember that we only need $n \le \ell \cdot |P|$ ($\le |\text{States}| \cdot |P|$) to get $\text{Rep}^\infty(P)$. Hence we have an algorithm for computing $\mathcal{S}^\infty(P)$ which runs in time $O(|\text{Tab}|^4 \cdot |\text{States}|^3 \cdot |P|)$. ◀ ◀

A.5 Proof of Lemma 13

▶ **Lemma 13.** *If $\langle S, E\rangle \in \mathsf{Sol}(L, W)$ and S is reachable from s in $\mathcal{S}^\infty(P(S))$, then it satisfies condition (‡). Conversely, if $\langle S, E\rangle$ satisfies condition (‡), then there exists $\langle S', E'\rangle \in \mathsf{Sol}(L, W)$ such that $\langle S, E\rangle \subseteq \langle S', E'\rangle$.*

Proof. ($\Rightarrow$) Let $\langle S, E\rangle \in \mathsf{Sol}(L, W)$ and $P = P(S)$. Obviously $\langle S, E\rangle \subseteq \mathcal{S}^\infty(P)$ and $\langle S, E\rangle$ is strongly connected. Moreover $\Omega_A \cap S = \emptyset \Leftrightarrow A \in P$ by definition of $P(S)$. Also, $P \cap W = \emptyset$ because $\mathsf{Sol}(L, W) \subseteq \{\langle S, E\rangle \mid P \cap W = \emptyset\}$. And finally $L \subseteq P$ because the arguments are decreasing and initially does not contain any Ω_A for $A \in L$.
($\Leftarrow$) Assume that $\langle S, E\rangle$ satisfies condition (‡). We show by induction on the size of $\langle S', E'\rangle$ that if $\langle S, E\rangle \subseteq \langle S', E'\rangle$ then $\exists \langle S'', E''\rangle \in \mathsf{SSG}(\langle S', E'\rangle)$ such that $\langle S, E\rangle \subseteq \langle S'', E''\rangle$.

The basic case is when $\langle S', E'\rangle = \langle S, E\rangle$ and is obvious since $\mathsf{SSG}(\langle S, E\rangle) = \{\langle S, E\rangle\}$ (this is due to conditions (4) and (3)).

First assume that $\langle S', E'\rangle \subseteq \mathcal{S}^\infty(P(S'))$ and that $\langle S', E'\rangle$ is strongly connected. Then, $\mathsf{SSG}(\langle S', E'\rangle) = \{\langle S', E'\rangle\}$ and we can choose $\langle S'', E''\rangle = \langle S', E'\rangle$ to get the expected result.

Then assume that $\langle S', E'\rangle$ is not strongly connected or $\langle S', E'\rangle$ is not a subset of $\mathcal{S}^\infty(P(S'))$. The transition system $\langle S, E\rangle$ is strongly connected since it satisfies (‡). Hence there is a strongly connected component of $\langle S', E'\rangle$, say $\langle S'', E''\rangle$, which contains $\langle S, E\rangle$. We have $P(S'') \subseteq P(S)$ hence $\mathcal{S}^\infty(P(S)) \subseteq \mathcal{S}^\infty(P(S''))$ and thus $\langle S, E\rangle \subseteq \langle S'', E''\rangle \cap \mathcal{S}^\infty(P(S''))$. The transition system $\langle S'', E''\rangle \cap \mathcal{S}^\infty(P(S''))$ is smaller than $\langle S', E'\rangle$ since either $\langle S'', E''\rangle \subsetneq \langle S', E'\rangle$ (in case $\langle S', E'\rangle$ is not strongly connected), or $\langle S'', E''\rangle \cap \mathcal{S}^\infty(P(S'')) \subseteq \langle S', E'\rangle \cap \mathcal{S}^\infty(P(S'')) \subseteq \langle S', E'\rangle \cap \mathcal{S}^\infty(P(S')) \subsetneq \langle S', E'\rangle$ (in case $\langle S', E'\rangle$ is not a subset of $\mathcal{S}^\infty(P(S'))$). We apply the induction hypothesis to the set $\langle S'', E''\rangle \cap \mathcal{S}^\infty(P(S''))$ and we get the expected result.

Now because of condition (1), $S \cap \bigcup_{A \in L} \Omega_A = \emptyset$. Hence, due to the previous analysis, there exists $\langle S', E'\rangle \in \mathsf{SSG}\left(\langle \text{States} \setminus \bigcup_{A \in L} \Omega_A, \text{Edg}\rangle\right)$ such that $\langle S, E\rangle \subseteq \langle S', E'\rangle$. Now because of condition (2), $P(S) \cap W = \emptyset$, hence $P(S') \cap W = \emptyset$. This concludes the proof of the lemma. ◀ ◀

A.6 Proof of Lemma 14

▶ **Lemma 14.** *The set $\mathsf{Sol}(L, W)$ can be computed in polynomial time.*

Proof. Each recursive call of SSG applies to a decomposition in SCCs of the current transition system under consideration. Hence the number of recursive calls is bounded by $|\text{States}|^2$. Computing the decomposition in SCCs can be done in linear time [A6]. Furthermore we have assumed that we could compute $\mathcal{S}^\infty(P)$ in polynomial time if $P \subseteq \text{Agt}$ is given. Hence globally we can compute $\mathsf{Sol}(L, W)$ in polynomial time. ◀ ◀

A.7 Proof of the upper bound in Theorem 11

We prove the PTIME upper bound for the constrained existence problem. We fix a set W of winning players and L of losing players and we compute the set $\mathsf{Sol}(L, W)$. Then for each

$\langle S, E\rangle$ in $\mathsf{Sol}(L, W)$, we check that S is reachable from s in $\mathcal{S}^\infty(P(S))$. If this is the case, we answer "yes, there is an equilibrium from s that satisfies the constraint", otherwise we answer "no, there is no equilibrium from s that satisfies the constraint". Applying the above results, this algorithm runs in polynomial time (Lemma 14 and Lemma 32). It remains to prove its correctness.

Soundness. Assume that the algorithm answers "yes" due to $\langle S, E\rangle$. Then by Lemma 13, $\langle S, E\rangle$ satisfies condition (‡). Then by the set-based characterisation (Corollary 12), we get that there is an infinite path ρ from s which lies in $\mathcal{S}^\infty(P(S))$, is losing for all players in L and winning for all players in W. Applying the result of the previous subsection (Proposition 9), there is some strategy profile σ_{Agt} such that $(\sigma_{\mathrm{Agt}}, \rho)$ is a pseudo-Nash equilibrium in $\mathcal{G}$ from s, and it satisfies the specified constraint on the sets of losing and winning players.

Completeness. Conversely assume $(\sigma_{\mathrm{Agt}}, \rho)$ is a pseudo-Nash equilibrium in $\mathcal{G}$ from s which satisfies the constraint on the sets of losing and winning players. Write P for the set of losing players for that equilibrium. From Proposition 9, we get that ρ is an infinite path in $\mathcal{S}^\infty(P)$ which ends up in some $\langle S, E\rangle$ with $P = P(S)$. From Corollary 12, $\langle S, E\rangle$ satisfies the condition (‡), which proves that there is some $\langle S', E'\rangle \in \mathsf{Sol}(L, W)$ such that $\langle S, E\rangle \subseteq \langle S', E'\rangle$. The set S is reachable from s in $\mathcal{S}^\infty(P(S))$, hence S' is also reachable from s in $\mathcal{S}^\infty(P(S))$, and as $P(S') \subseteq P(S)$, we get that $\mathcal{S}^\infty(P(S)) \subseteq \mathcal{S}^\infty(P(S'))$ and we deduce that S' is reachable from s in $\mathcal{S}^\infty(P(S'))$. Hence the algorithm will answer "yes". ◀

B Appendix for Section 4

B.1 Proof of Proposition 16

▶ **Proposition 16.** *Any game simulation is transitive.*

Proof. We assume that $\mathcal{G}_2$ simulates $\mathcal{G}_1$ and that $\mathcal{G}_3$ simulates $\mathcal{G}_2$, and we write $\lhd_{12}$ and $\lhd_{23}$ for the respective game-simulation relations. We define the relation $\lhd$ between $\mathcal{G}_1$ and $\mathcal{G}_3$ by: $s_1 \lhd s_3$ iff there exists s_2 such that $s_1 \lhd_{12} s_2$ and $s_2 \lhd_{23} s_3$. We show that $\lhd$ is a game simulation relation.

Assume $s_1 \lhd s_3$ and fix s_2 such that $s_1 \lhd_{12} s_2$ and $s_2 \lhd_{23} s_3$. Fix some move m^1_{Agt} in $\mathcal{G}_1$, and apply the transfer property to $\lhd_{12}$. This gives a move m^2_{Agt} in $\mathcal{G}$ which satisfies the two conditions (a) and (b). We can apply once more transfer property to $\lhd_{23}$ and m^2_{Agt}, and we get a move m^3_{Agt} which satisfies the two conditions (a) and (b).

1. Fix some state s'_3 in $\mathcal{G}_3$. There exists some state s'_2 in $\mathcal{G}_2$ such that $s'_2 \lhd_{23} s'_3$ and $\mathrm{Susp}((s_2, s'_2), m^2_{\mathrm{Agt}}) \subseteq \mathrm{Susp}((s_3, s'_3), m^3_{\mathrm{Agt}})$. Also there exists some state s'_1 in $\mathcal{G}_1$ such that $s'_1 \lhd_{23} s'_2$ and $\mathrm{Susp}((s_1, s'_1), m^1_{\mathrm{Agt}}) \subseteq \mathrm{Susp}((s_2, s'_2), m^2_{\mathrm{Agt}})$. We thus get $s_1 \lhd s_3$ and $\mathrm{Susp}((s_1, s'_1), m^1_{\mathrm{Agt}}) \subseteq \mathrm{Susp}((s_3, s'_3), m^3_{\mathrm{Agt}})$.
2. Fix $(s_1, s'_1) \in \mathrm{Tab}_1(s_1, m^1_{\mathrm{Agt}})$. There exists some $(s_2, s'_2) \in \mathrm{Tab}_2(s_2, m^2_{\mathrm{Agt}})$ such that $s'_1 \lhd_{12} s'_2$. Then, there exists some $(s_3, s'_3) \in \mathrm{Tab}_3(s_3, m^3_{\mathrm{Agt}})$ such that $s'_2 \lhd_{23} s'_3$. Hence picking this precise (s_3, s'_3), we get that $s'_1 \lhd s'_3$.

This implies the transfer property that characterises game simulations: $\lhd$ is a game simulation relation. ◀ ◀

B.2 Proof of Proposition 17

▶ **Proposition 17.** *Assume $\mathcal{G}$ and $\mathcal{G}'$ are games. Fix two states s and s' in $\mathcal{G}$ and $\mathcal{G}'$ respectively, and assume that $\lhd$ is a winning-preserving (from (s, s')) game simulation. If* *there exists a pseudo-Nash equilibrium $(\sigma_{\mathrm{Agt}},\rho)$ of $\mathcal{G}$ from s, then there exists a pseudo-Nash equilibrium $(\sigma'_{\mathrm{Agt}},\rho')$ of $\mathcal{G}'$ from s' with $\rho \lhd \rho'$. In particular, ρ and ρ' have the same payoff.*

Proof. We fix a strategy profile σ_{Agt} in $\mathcal{G}$ and $\rho \in \mathrm{Out}_{\mathcal{G}}(s,\sigma_{\mathrm{Agt}})$. We will construct a strategy profile σ'_{Agt} in $\mathcal{G}'$ and a path $\rho' \in \mathrm{Out}_{\mathcal{G}'}(s',\sigma'_{\mathrm{Agt}})$ such that:

(a) for every $\overline{\rho}' \in \mathrm{Play}_{\mathcal{G}'}(s')$, either $\mathrm{Susp}(\overline{\rho}',\sigma'_{\mathrm{Agt}}) = \emptyset$, or there exists $\overline{\rho} \in \mathrm{Play}_{\mathcal{G}}(s)$ s.t. $\overline{\rho} \lhd \overline{\rho}'$ and $\mathrm{Susp}(\overline{\rho}',\sigma'_{\mathrm{Agt}}) \subseteq \mathrm{Susp}(\overline{\rho},\sigma_{\mathrm{Agt}})$;
(b) $\rho \lhd \rho'$.

Assume we have done the construction, and that $(\sigma_{\mathrm{Agt}},\rho)$ is a pseudo-Nash equilibrium in $\mathcal{G}$. We want to prove that $(\sigma'_{\mathrm{Agt}},\rho')$ is a pseudo-Nash equilibrium in $\mathcal{G}'$. First notice that the payoff of ρ and ρ' are the same (because $\rho \lhd \rho'$ and $\lhd$ is winning-preserving from (s,s')), say ν. Towards a contradiction assume that there is a player B such that $\nu_B = 0$, and a B-strategy σ' in $\mathcal{G}'$ such that there is $\overline{\rho}' \in \mathrm{Out}_{\mathcal{G}'}(s',\sigma'_{\mathrm{Agt}}[B \mapsto \sigma'])$ and $\overline{\rho}'$ is winning for player B. Note that $B \in \mathrm{Susp}(\overline{\rho}',\sigma'_{\mathrm{Agt}})$. Applying (a) above, there exists $\overline{\rho} \in \mathrm{Play}_{\mathcal{G}}(s)$ such that $\overline{\rho} \lhd \overline{\rho}'$ and $\mathrm{Susp}(\overline{\rho}',\sigma'_{\mathrm{Agt}}) \subseteq \mathrm{Susp}(\overline{\rho},\sigma_{\mathrm{Agt}})$. In particular, $B \in \mathrm{Susp}(\overline{\rho},\sigma_{\mathrm{Agt}})$, and there exists a B-strategy σ such that $\overline{\rho} \in \mathrm{Out}_{\mathcal{G}}(s,\sigma_{\mathrm{Agt}}[B \mapsto \sigma])$. As $\overline{\rho} \lhd \overline{\rho}'$ and $\lhd$ is winning-preserving from (s,s'), $\overline{\rho}$ is winning for Player B, which contradicts the fact that $(\sigma_{\mathrm{Agt}},\rho)$ is a pseudo-Nash equilibrium. Hence, we have proven that $(\sigma'_{\mathrm{Agt}},\rho')$ is a pseudo-Nash equilibrium in $\mathcal{G}'$ from s'.

It remains to show how we construct $(\sigma'_{\mathrm{Agt}},\rho')$. We first build inductively ρ' and define σ'_{Agt} along that path. By induction on i, we build a path $\rho'_{\leq i}$ and move $\sigma'_{\mathrm{Agt}}(\rho'_{\leq i})$ such that $\rho_{\leq i} \lhd \rho'_{\leq i}$ and $\sigma'_{\mathrm{Agt}}(\rho'_{\leq i})$ is linked to $\sigma_{\mathrm{Agt}}(\rho_{\leq i})$ through the transfer property of $\lhd$.

- Case $i = 0$: we have $\rho_{=0} = s \lhd s' = \rho'_{=0}$, and we define $\sigma'_{\mathrm{Agt}}(\rho'_{=0})$ as given by the transfer property for $\sigma_{\mathrm{Agt}}(\rho_{=0})$.
- Inductive step: Assume we have built $\rho'_{\leq i}$ and $\sigma'_{\mathrm{Agt}}(\rho'_{\leq i})$ which satisfy the induction hypothesis. As $\sigma'_{\mathrm{Agt}}(\rho'_{\leq i})$ is linked to $\sigma_{\mathrm{Agt}}(\rho_{\leq i})$ through the transfer property, we apply condition (2) to $(\rho_{=i},\rho_{=i+1}) \in \mathrm{Tab}(\rho_{=i},\sigma_{\mathrm{Agt}}(\rho_{\leq i}))$, and we get a state $\rho'_{=i+1}$ in $\mathcal{G}'$ such that $(\rho'_{=i},\rho'_{=i+1}) \in \mathrm{Tab}(\rho'_{=i},\sigma_{\mathrm{Agt}}(\rho'_{\leq i}))$ and $\rho_{=i+1} \lhd \rho'_{=i+1}$. We can apply the transfer property to move $\sigma_{\mathrm{Agt}}(\rho_{\leq i+1})$ and we get a move that we feed in $\sigma'_{\mathrm{Agt}}(\rho'_{\leq i+1})$. This proves the induction hypothesis.

So far we have constructed a path ρ' in $\mathcal{G}'$ and a strategy profile σ'_{Agt} over all prefixes of ρ', and it is the case that $\rho' \in \mathrm{Out}_{\mathcal{G}'}(s',\sigma'_{\mathrm{Agt}})$. We will do so by defining $\sigma'_{\mathrm{Agt}}(h')$ by induction on the length of the history h', and we will make sure that for every $h' \in \mathrm{Hist}_{\mathcal{G}'}(s')$, either $\mathrm{Susp}(h',\sigma'_{\mathrm{Agt}}) = \emptyset$, or there exists $h \in \mathrm{Hist}_{\mathcal{G}}(s)$ such that $h \lhd h'$, $\sigma'_{\mathrm{Agt}}(h')$ is linked to $\sigma_{\mathrm{Agt}}(h)$ through the transfer condition, and $\mathrm{Susp}(h',\sigma'_{\mathrm{Agt}}) \subseteq \mathrm{Susp}(h,\sigma_{\mathrm{Agt}})$.

The base case is when h' is a prefix of ρ' (this is the case of the 0-length history s'), and the expected property is satisfied. Assume prefix we have done a proper construction for history h', and we fix a new history $h' \cdot t' \in \mathrm{Hist}_{\mathcal{G}'}(s')$. By induction hypothesis, we have that either $\mathrm{Susp}(h',\sigma'_{\mathrm{Agt}}) = \emptyset$, or there is $h \in \mathrm{Hist}_{\mathcal{G}}(s)$ such that $h \lhd h'$, $\sigma'_{\mathrm{Agt}}(h')$ is linked to $\sigma_{\mathrm{Agt}}(h)$ through the transfer property, and $\mathrm{Susp}(h',\sigma'_{\mathrm{Agt}}) \subseteq \mathrm{Susp}(h,\sigma_{\mathrm{Agt}})$.

- In the first case we set $\sigma'_{\mathrm{Agt}}(h' \cdot t')$ to any move, and we are done.
- In the second case, due to the transfer property, there exists t in $\mathcal{G}$ such that $t \lhd t'$ and $\mathrm{Susp}((\mathrm{last}(h'),t'),\sigma'_{\mathrm{Agt}}(h')) \subseteq \mathrm{Susp}((\mathrm{last}(h),t),\sigma_{\mathrm{Agt}}(h))$. If $\mathrm{Susp}((\mathrm{last}(h'),t'),\sigma'_{\mathrm{Agt}}(h'))$ is empty, then we set $\sigma'_{\mathrm{Agt}}(h' \cdot t')$ to any move, and we are done since $\mathrm{Susp}(h' \cdot t',\sigma'_{\mathrm{Agt}}) = \emptyset$ as well. Otherwise, we get that $\mathrm{Susp}((\mathrm{last}(h),t),\sigma_{\mathrm{Agt}}(h)) \neq \emptyset$, hence $h \cdot t$ is a proper history in $\mathcal{G}$ which then satisfies $(h \cdot t) \lhd (h' \cdot t')$. We then apply the transfer property to t

and t' with $\sigma_{\mathrm{Agt}}(h \cdot t)$ as a move, and this yields a move m'_{Agt} that we put in $\sigma'_{\mathrm{Agt}}(h' \cdot t')$. Furthermore we get from above inclusions that $\mathrm{Susp}(h' \cdot t', \sigma'_{\mathrm{Agt}}) \subseteq \mathrm{Susp}(h \cdot t, \sigma_{\mathrm{Agt}})$.

This concludes the induction step, and the proof of the proposition. ◀ ◀

B.3 Proof of Lemma 18

▶ **Lemma 18.** *$\mathcal{G} \ltimes \mathcal{A}$ simulates $\mathcal{G}$, and vice versa. Furthermore, in both cases we can exhibit a game-simulation that is winning-preserving from any $(s,(s,q_0))$ (with $s \in \mathrm{States}$).*

Proof. We define the relations $\lhd$ and $\lhd'$ by $s \lhd (s,q)$, and $(s,q) \lhd' s$ for every $s \in \mathrm{States}$ and $q \in Q$. We show that $\lhd$ and $\lhd'$ are game-simulation relations.

First notice that if $((s_n,q_n))_{n\geq 0}$ is a play in $\mathcal{G} \ltimes \mathcal{A}$, then its π-projection $(s_n)_{n\geq 0}$ is a play in $\mathcal{G}$. Conversely, if $\rho = (s_n)_{n\geq 0}$ is a play in $\mathcal{G}$, then there is a unique path $(q_n)_{n\geq 0}$ from initial state q_0 in $\mathcal{A}$ which reads it, and $((s_n,q_n))_{n\geq 0}$ is then a path in $\mathcal{G} \ltimes \mathcal{A}$ that we write $\pi^{-1}(\rho) = ((s_n,q_n))_{n\geq 0}$. That way π defines a one-to-one correspondence between plays in $\mathcal{G}$ and plays in $\mathcal{G} \ltimes \mathcal{A}$ where the second component starts in q_0. For a player $A_i \neq A$, the objective is defined so that $\pi(\rho)$ has the same payoff as ρ. Consider now player A. The payoff is 1 in $\mathcal{G}$ for $\rho = (s_n)_{n\geq 0}$ iff $(s_n)_{n\geq 0} \in \mathrm{L}(\mathcal{A})$ iff the unique path $(q_n)_{n\geq 0}$ from initial state q_0 that reads $(s_n)_{n\geq 0}$ in $\mathcal{A}$ visits R infinitely often iff $\pi^{-1}(\rho)$ visits Ω infinitely often in $\mathcal{G} \ltimes \mathcal{A}$. This proves that $\lhd$ is winning-preserving.

It remains to show the transfer property for a game-simulation relation.

Assume $s \lhd (s,q)$ and pick a move m_{Agt} in $\mathcal{G}$. It is also a move in $\mathcal{G} \ltimes \mathcal{A}$. Take $(s',q') \in \mathrm{States}'$. Then by definition $s' \lhd (s',q')$.

- If $((s,q),(s',q'))$ is not an edge in $\mathcal{G} \ltimes \mathcal{A}$, then $\mathrm{Susp}(((s,q),(s',q')), m_{\mathrm{Agt}}) = \emptyset$, and condition (1) trivially holds.
- If $((s,q),(s',q'))$ is an edge in $\mathcal{G} \ltimes \mathcal{A}$, then this means that $\delta(q,s) = q'$. For any move m'_{Agt} in $\mathcal{G}$, we have that $(s,s') \in \mathrm{Tab}(s, m'_{\mathrm{Agt}})$ iff $((s,q),(s',\delta(q,s))) \in \mathrm{Tab}'((s,q), m'_{\mathrm{Agt}})$. Therefore $\mathrm{Susp}(((s,q),(s',q')), m_{\mathrm{Agt}}) = \mathrm{Susp}((s,s'), m_{\mathrm{Agt}})$, which implies condition (1).

Condition (2) obviously holds since , $(s,s') \in \mathrm{Tab}(s, m_{\mathrm{Agt}})$ if and only if $((s,q),(s',\delta(q,s))) \in \mathrm{Tab}'((s,q), m_{\mathrm{Agt}})$ by definition of $\mathcal{G} \ltimes \mathcal{A}$.

We now assume $(s,q) \lhd' s$ and pick a move m_{Agt} in $\mathcal{G} \ltimes \mathcal{A}$. It is also a move in $\mathcal{G}$. Take $s' \in \mathrm{States}$. We define $q' = \delta(q,s)$, and we have $(s',q') \lhd s'$ by definition of $\lhd'$. With the same case distinction as before, we get that either $\mathrm{Susp}((s,s'), m_{\mathrm{Agt}}) = \emptyset$ or $\mathrm{Susp}(((s,q),(s',q')), m_{\mathrm{Agt}}) = \mathrm{Susp}((s,s'), m_{\mathrm{Agt}})$. This implies condition (1). As before condition (2) obviously holds. ◀ ◀

B.4 Proof of Corollary 19

▶ **Corollary 19.** *Let $s \in \mathrm{States}$ and $\nu\colon \mathrm{Agt} \to \{0,1\}$. There is a pseudo-Nash equilibrium $(\sigma_{\mathrm{Agt}}, \rho)$ in $\mathcal{G}$ from s with payoff ν if and only if there is a pseudo-Nash equilibrium $(\sigma'_{\mathrm{Agt}}, \rho')$ in $\mathcal{G}'$ from $(s, q_{01}, \dots, q_{0n})$ with payoff ν, where q_{0i} is the initial state of $\mathcal{A}_i$.*

Proof. By Lemma 18 and the transitivity of game simulations (Proposition 16), the games $\mathcal{G}$ and $\mathcal{G}'$ simulate each other, with winning-preserving simulations from $(s,(s,q_{01},\dots,q_{0n}))$. The claim now follows from Proposition 17. ◀ ◀

C Appendix for Section 5

C.1 Proof of Theorem 21

▶ **Theorem 21.** *The verification, existence, and constrained existence problems for finite games with objectives defined by deterministic Büchi automata are* EXPTIME*-complete.*

Proof. Let $\mathcal{C} = (S, T)$ be a *weighted automaton* (made of a finite set S of states and a finite set of transitions $T \subseteq S \times (\mathbb{N} \setminus \{0\}) \times S$), and $c \in \mathbb{N}$. The countdown game on $\mathcal{C}$ is played as follows: from some state s of S, Player 1 selects an integer d such that $(s, d, s') \in T$ for some s'. Player 2 then selects one of the states s' for which $(s, d, s') \in T$. A counter b is increased by d, and the game continues from s', until the counter reaches or exceeds c. Player 1 wins if, starting with $b = 0$, the counter reaches exactly c at some point. Player 2 wins otherwise.

An instance of the countdown game is encoded into a new game $\mathcal{G}$ as follows: besides the two players of the countdown game, we also have $n+2$ extra players, where $n = \lceil \log_2(c+1) \rceil$. The first $n+1$ extra players, named B_0 to B_n, are responsible for storing the $n+1$ bits of the counter[3] The last extra player B_r is in charge of propagating the possible carries involved when performing the addition of d to the counter. The deterministic Büchi automata defining the objectives of these $n+2$ extra players is depicted on Fig. 4 (where States is the set of states of the new game $\mathcal{G}$). In the automaton defining the objective of Player B_i, there are four important actions: two "assignment" actions $b_i \leftarrow 0$ and $b_i \leftarrow 1$, and two "test" actions $b_i = 0$ and $b_i = 1$. In order to win, Player B_i cannot play action $b_i = 0$ if the last assignment was $b_i \leftarrow 1$, and symmetrically she will not play action $b_i = 1$ after assignment $b_i \leftarrow 0$.

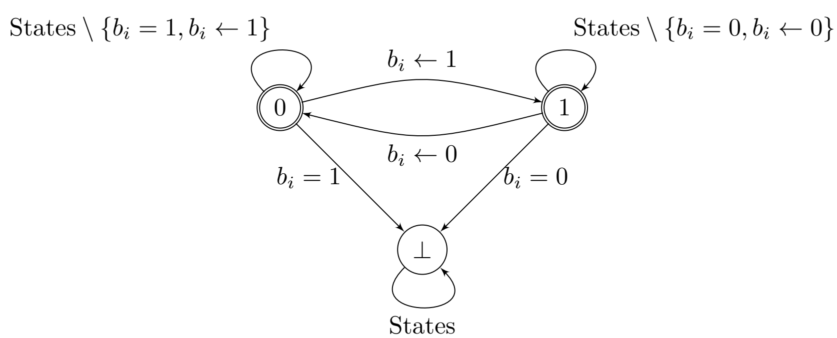

Figure 4 Büchi automata describing the objectives of player B_i

Each transition (s, d, s') in the weighted game (assuming the binary representation of d is $d_n d_{n-1} \cdots d_0$, with $d_n = 0$ since $d \leq c$) is then replaced with a sequence of modules, as depicted on Fig. 5. In that sequence, nodes labelled with s and s' are states of the game,

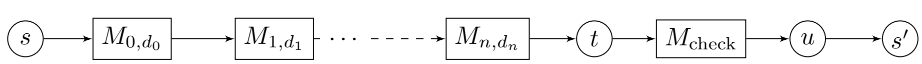

Figure 5 Encoding a transition (s, d, s')

[3] As a first step of the encoding, we remove transitions whose weight is larger than c, as they are useless to Player 1. We can then assume that b is always less than $2c - 1$, hence it can be encoded on $n+1$ bits.

corresponding to the states with the same name in the original countdown game. States t and u are extra states, visited just before and just after M_{check}. Modules M_{i,d_i} and M_{check} are gadgets, used respectively to perform addition of bit d_i at position i of the counter, and to check whether counter b reaches or exceeds the target value c. Module M_{i,d_i} is depicted on Fig. 6; actually, the module is depicted in the case where $d_i = 0$. The module for $d_i = 1$ is obtained by flipping the first transition to state $d_i = 1$.

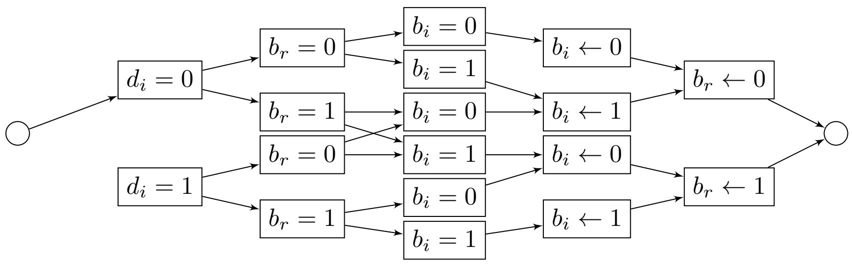

Figure 6 Module M_{i,d_i}

States $d_i = 0$ and $d_i = 1$ are controlled by Player B_r, in charge of the carry. In case the addition on the previous bit has generated a carry, Player B_r has to go to state $b_r = 1$, and otherwise she has to go to $b_r = 0$. States labelled with $b_r = 0$ and $b_r = 1$ of module M_{i,d_i} are controlled by Player B_i. As above, if at the previous step of the game the bit b_i of the counter was set to 0, then Player B_i has to play to $b_i = 0$, and she will play to $b_i = 1$ otherwise. Playing differently would be losing. The subsequent states are used to update the values of i-th bit b_i of the counter, and the carry b_r. It can be checked that when starting from configuration $(s, b_r = 0, b_0, \cdots, b_n)$ (i.e., from state s of $\mathcal{G}$, with automaton B_r in state 0 and automata B_i in state b_i), we reach configuration $(t, b'_r, b'_0, \cdots, b'_n)$ with

- either one (at least) of b'_r and b'_0 to b'_n is $\perp$,
- or $b'_r = 0$ and, writing b and b' for the integers represented by $b_n b_{n-1} \cdots b_0$ and $b'_n b'_{n-1} \cdots b'_0$ resp., it holds $b' = b + d$.

Let us now focus on M_{check}, depicted on Fig. 7. The role of this module is to decide which of $b < c$, $b = c$ or $b > c$, is the case. Nodes t and u are the states with the same name

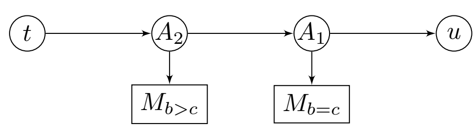

Figure 7 Module M_{check}

on Fig. 5. Nodes A_2 and A_1 are controlled by Player 2 and Player 1, respectively. Modules $M_{b>c}$ and $M_{b=c}$ are used to compare b to c, and decide whether the game is over (and then who won) or if it should go on. Module $M_{b>c}$ leads to the winning state W_2 of Player 2 if it is indeed the case that $b > c$, where b is the value represented by the states of the Büchi automata B_0 to B_n, and to the winning state W_1 of Player 1 otherwise. Symmetrically, module $M_{b=c}$ leads to the winning state of Player 1 if indeed $b = c$, and to the winning state of Player 2 otherwise. The modules $M_{b=c}$ is depicted on Fig. 8, where for all $i \in [0, n]$, Player B_i controls the state immediately preceding the state labelled with $b_i = c_i$. The module $M_{b>c}$ can be built similarly.

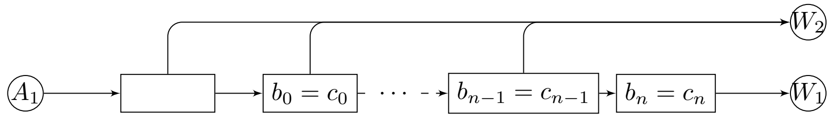

Figure 8 Module $M_{b=c}$

To conclude the reduction, in order to encode the fact that Player 2 chooses one of the possible successors once Player 1 has selected d, we add transitions from state u of the module of Fig. 5 to all possible states s' for which $(s, d, s') \in T$.

Our construction encodes the original countdown game with the following correspondence:

▶ **Lemma.** *There is a winning strategy for Player 1 in the countdown game $\mathcal{C}$ if and only if there is a Nash equilibrium in the game $\mathcal{G}$ where all players but Player 2 win.*

As a consequence, the verification- and constrained existence problem are EXPTIME-hard. In order to get the result for the existence problem, we modify our construction with a small initial module, as depicted on Fig. 9. The new game $\mathcal{G}'$ is obtained by adding two copies of

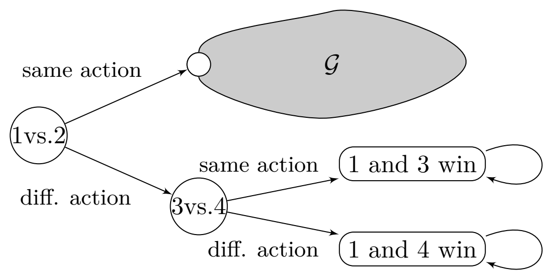

Figure 9 Game $\mathcal{G}'$ for proving EXPTIME-hardness for the existence problem

the matching-penny game: in the initial state, Players 1 and 2 (the same as in $\mathcal{G}$) have two possible actions. The play proceeds to $\mathcal{G}$ if they propose the same action. Otherwise, the play enters another matching-penny game between two new players (Players 3 and 4). Again, they have two possible actions, and Player 3 wins if they play the same action; otherwise Player 4 wins. In both cases, Player 1 is also declared the winner. These winning conditions can obviously be encoded as Büchi automata.

In the lower subgame, there is no Nash equilibrium. Hence if there is a Nash equilibrium in $\mathcal{G}'$, it goes in $\mathcal{G}$. Moreover, if Player 1 is not winning, then she can change her action in the first state and win, hence the original strategy was not a Nash equilibrium. It follows that if there is a Nash equilibrium in the game of Fig. 9, it visits $\mathcal{G}$ and is winning for Player 1. It is easily checked that the $n + 2$ extra players always have a winning strategy in $\mathcal{G}$, and that Player 2 will not win if Player 1 does. The existence of a Nash equilibrium in $\mathcal{G}'$ thus implies the existence of a Nash equilibrium where all but Player 2 win in $\mathcal{G}$. Conversely, if there is such an equilibrium in $\mathcal{G}$, then there is one in $\mathcal{G}$'. This concludes the proof. ◀ ◀

C.2 Proof of Theorem 22

▶ **Theorem 22.** *The verification, existence, and constrained existence problems for finite games with objectives defined by* 1*-weak deterministic Büchi automata are* PSPACE*-complete.*

We start by proving the upper bound.

▶ **Proposition.** *The verification, existence, and constrained existence problems for finite games with objectives defined by* 1*-weak deterministic Büchi automata are in* PSPACE.

Proof. In order to fix notations, assume that each player A_i has its objective specified by a deterministic Büchi automaton $\mathcal{A}_i = \langle Q_i, \mathrm{States}, \delta_i, q_{i,0}, R_i\rangle$, where $Q_i = \{q_{i,0}, q_{i,1}, \ldots, q_{i,n_i}\}$. We assume now that each strongly-connected components of the automaton has only one state. We can then define a partial order on each set Q_i, given by the transitions of $\mathcal{A}_i$. We extend this order to the product $\mathcal{A}$ of these automata:

$$(q_{A_1}, \ldots, q_{A_p}) > (q'_{A_1}, \ldots, q'_{A_p}) \quad \Leftrightarrow \quad \forall i \in \mathrm{Agt}.\ q_i \geq q'_i \ \wedge\ \exists j \in \mathrm{Agt}.\ q_j > q'_j.$$

In the algorithm, we write $q\xrightarrow{s}$ for the successor of q by s in the product $\prod_{i\in\mathrm{Agt}} \mathcal{A}_i$ (which is deterministic). Notice that $\mathrm{Susp}_{\mathcal{G}}((s,s'), m_{\mathrm{Agt}})$ is equal to $\mathrm{Susp}_{\mathcal{H}}(((s,q),(s',q\xrightarrow{s})), m_{\mathrm{Agt}})$ for all q, and that $\mathrm{Susp}_{\mathcal{H}}(((s,q),(s',q')), m_{\mathrm{Agt}}) = \varnothing$ when $q' \neq q\xrightarrow{s}$.

Termination. The algorithm terminates since the recursive calls $f(P',n',q')$ (lines 16 and 20) in the computation of $f(P,n,q)$ are made with either $P' \subsetneq P$ or $n' < n$ or $q' < q$.

Correctness. We prove that $s \in f(P,n,q)$ iff $(s,q) \in \mathrm{Rep}^n_{\mathcal{H}}(P)$. The proof proceeds by induction. The base cases ($P = \varnothing$ and $n = 0$) are trivial.

When $P \neq \varnothing$ or $n > 0$, assume that the procedure is correct for all $P' \subsetneq P$ and all $n' < n$. We prove that, for all q, $\{s \mid (s,q) \in \mathrm{Rep}^n(P)\} \subseteq S'$ holds at each step of the **repeat** loop.

It is true at the beginning since $S' = \mathrm{States}$. Now, let s s.t. $(s,q) \in \mathrm{Rep}^n_{\mathcal{H}}(P)$. We prove that this state will not be removed from S', i.e., that the condition on line 31 will always evaluate to false. From the definition of the repellor, there is a move m_{Agt} s.t. for all (s',q'), it holds $(s',q') \in \mathrm{Rep}^n_{\mathcal{H}}(P')$ and if $(s',q') \in \Omega_B$ for some $B \in P'$ then $(s',q') \in \mathrm{Rep}^{n-1}_{\mathcal{H}}(P')$, where $P' = P \cap \mathrm{Susp}(((s,q),(s',q')), m_{\mathrm{Agt}})$. We pick such an m_{Agt}, which we prove will set `found_m` to $\top$.

Pick any $s' \in \mathrm{States}$ and let $P' = P \cap \mathrm{Susp}(((s,q),(s',q\xrightarrow{s})), m_{\mathrm{Agt}})$. We have to prove that `found_s'` is not set to $\top$. By choice of m_{Agt}, for all $B \in P'$, we have $(s,q\xrightarrow{s}) \notin \Omega_B \setminus \mathrm{Rep}^{n-1}_{\mathcal{H}}(P')$, so that either $(s',q\xrightarrow{s}) \in \mathrm{Rep}^{n-1}_{\mathcal{H}}(P')$, or $(s,q\xrightarrow{s}) \notin \Omega_B$ for all $B \in P'$. By induction hypothesis, the condition on line 16 is evaluated to false.

We now reason by induction on q: if q is the smallest state (i.e., $q \xrightarrow{s} q$ for all $s \in \mathrm{States}$), then

- either $P \subseteq \mathrm{Susp}(((s,q),(s',q\xrightarrow{s})), m_{\mathrm{Agt}})$, and from the choice of m_{Agt}, we have

$$(s',q\xrightarrow{s}) = (s',q) \in \mathrm{Rep}^n_{\mathcal{H}}(P \cap \mathrm{Susp}(((s,q),(s',q\xrightarrow{s})), m_{\mathrm{Agt}})) = \mathrm{Rep}^n_{\mathcal{H}}(P),$$

  so that $s' \in S'$, and the condition on line 20 is evaluated to false.
- or $P \cap \mathrm{Susp}(((s,q),(s',q\xrightarrow{s})), m_{\mathrm{Agt}}) \subsetneq P$, and, again by choice of m_{Agt}, we have

$$(s',q\xrightarrow{s}) \in \mathrm{Rep}^n_{\mathcal{H}}(P \cap \mathrm{Susp}(((s,q),(s',q\xrightarrow{s})), m_{\mathrm{Agt}})).$$

  Since $P \cap \mathrm{Susp}(((s,q),(s',q\xrightarrow{s}))$ is a strict subset of P, the induction hypothesis entails that $s' \in f(P \cap \mathrm{Susp}(((s,q),(s',q\xrightarrow{s})), m_{\mathrm{Agt}}), n, q\xrightarrow{s})$. Again, the condition on line 20 is evaluated to false.

Now, we assume that the result holds for $q\xrightarrow{s}$, and prove that it holds for q:

- if $P \subseteq \mathrm{Susp}(((s,q),(s',q\xrightarrow{s})), m_{\mathrm{Agt}})$ and $q \xrightarrow{s} q$, then with the same arguments as above, (s',q) is in $\mathrm{Rep}^n_{\mathcal{H}}(P)$, so that $s' \in S'$, and the condition on line 20 is evaluated to false.

Algorithm 1 Procedure $f(P, n, q)$

```
if n = 0 then
  return ∅
end if
if P = ∅ then
  return States
end if
S' ← States
repeat
  S ← S'
  for all s ∈ S do
    found_m = ⊥
    for all m_Agt ∈ Mov(s) do
      found_s' = ⊥
      for all s' ∈ States do
        P' ← Susp(s, s', m_Agt)
        if s' ∉ f(P ∩ P', n − 1, q -s->) and ∃B ∈ P ∩ P'. (s', q -s->) ∈ Ω_B then
          found_s' = ⊤
          break
        end if
        if ( ((P ⊆ P' and q = q -s->) and s' ∉ S')
             or
             ((P ⊈ P' or q ≠ q -s->) and s' ∉ f(P' ∩ P, n, q -s->)) ) then
          found_s' = ⊤
          break
        end if
      end for
      if ¬found_s' then
        found_m = ⊤
        break
      end if
    end for
  end for
  if ¬found_m then
    S' ← S' \ {s}
  end if
until S' = S
return S'
```

- Otherwise, either $P \cap \mathrm{Susp}(((s,q),(s',q\xrightarrow{s})),m_{\mathrm{Agt}}) \subsetneq P$, or $q \neq q\xrightarrow{s}$. In both cases, $(s',q\xrightarrow{s}) \in \mathrm{Rep}^n_{\mathcal{H}}(P \cap \mathrm{Susp}(((s,q),(s',q\xrightarrow{s})),m_{\mathrm{Agt}}))$, and applying the induction hypothesis (on P or on q, resp.), and the condition on line 20 is again evaluated negatively.

When the fixpoint is reached ($S = S'$ on line 34), S' satisfies the condition defining $\mathrm{Rep}^n(P)$ and therefore is included in it.

Space usage. If we omit the nested calls to f, the space needed to compute $f(P,n,q)$ is $O(|\mathrm{States}| + |\mathrm{Agt}| \cdot \log|\mathrm{Act}| + |\mathrm{Agt}| + n + \sum_{A_i \in \mathrm{Agt}} \log|Q_i| + m + m')$, where m is the memory needed to compute $q \xrightarrow{s}$, and m' is the memory needed determine whether (q,s) is in Ω_B. In the case where the automaton is represented explicitly m and m' are constant. Now, the maximal number of nested calls to f is $|P| + n + \sum_{A_i \in \mathrm{Agt}} \ell_i$, where ℓ_i is the length of the longest acyclic path in $\mathcal{A}_i$. Finally, according to Lemma 8, n can be bounded by $|\mathrm{States}| \cdot (\sum_{i \in \mathrm{Agt}} \ell_i) \cdot |P|$. The whole computation then runs in polynomial space in the case where the Büchi automata are represented explicitly. If they are not, it is enough to have that m, m' and ℓ_i can be polynomialy bounded to obtain a polynomial space algorithm. ◀ ◀

The lower bound holds already for the verification problem in safety games.

▶ **Proposition.** *The verification problem for finite safety games is* PSPACE*-hard.*

Proof. We proceed by a reduction from QSAT, the satisfiability problem for quantified Boolean formulae. Let ϕ be a quantified Boolean formula, of the form

$$\phi = Q_1 x_1.\ Q_2 x_2.\ \ldots\ Q_n x_n.\ C_1 \wedge C_2 \wedge \cdots \wedge C_m$$

where for all i, $Q_i \in \{\forall, \exists\}$, and for all j, C_j is either $\perp$ or a non-empty disjunction $\bigvee_k \ell_{j,k}$ of literals from $\{x_i, \neg x_i \mid 1 \le i \le n\}$ (we assume $n \ge 1$).

With ϕ, we associate a nondeterministic turn-based game $\mathcal{G}(\phi)$ with safety objectives. The set of states is $\mathrm{States} = \{s_i, x_i, \neg x_i \mid 1 \le i \le n\} \cup \{s_{n+1}\}$, with transitions $\mathrm{Edg} = \{(s_i,x_i),(s_i,\neg x_i),(x_i,s_{i+1}),(\neg x_i,s_{i+1}) \mid 1 \le i \le n\} \cup \{(s_{n+1},s_{n+1})\}$. The agents are $\mathrm{Agt} = \{A\} \cup \{C_j \mid 1 \le j \le m\}$. The states s_i for which $Q_i = \exists$ are controlled by Player A. The states s_i for which $Q_i = \forall$ have nondeterministic transitions to either x_i or $\neg x_i$. An example of this construction is depicted on Fig. 10. The winning condition for A is always false (i.e., A always loses), while the safety condition for Player C_j is $\Omega_{C_j} = \{\ell_{j,k} \mid k\}$ (Player C_j wins if none of these states are visited).

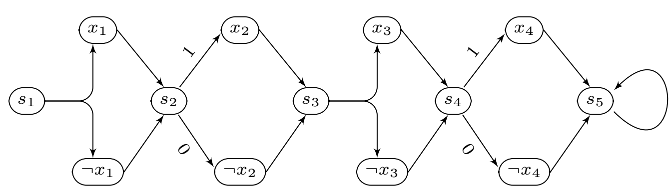

Figure 10 Example of the construction for $\phi = \forall x_1.\ \exists x_2.\ \forall x_3.\ \exists x_4.\ C_1 \wedge \cdots \wedge C_m$

▶ **Lemma.** *ϕ is valid iff Player A has a strategy under which all the players C_j lose along all the outcomes.*

Proof. We begin with the direct implication, by induction on n. For the base case, $\phi = Q_1 x_1 \bigwedge_j C_j$ where C_j only involve x_1, we consider two cases:

- $Q_1 = \exists$: since we assume ϕ is true, there must exist a value for x_1 which makes all clauses true. If this value is $\top$, consider the strategy $\sigma_\top$ of Player A such that $\sigma_\top(s_1) = x_1$. Assume that Player C_j wins under this strategy. This means that none of the literals that appear in C_j are visited, which implies that C_j only contains the literal $\neg x_1$. This contradicts the fact that $x_1 = \top$ makes all clauses true. The same argument applies if the value for x_1 was $\bot$.
- if $Q_1 = \forall$: then Player A has only one strategy. Again, assume that some player C_j wins along some outcome: as previously, this implies that for some value of x_1, the corresponding clause evaluates to false, which contradicts that ϕ is valid.

Now, assume that the result holds for all QSAT-instances with at most $n-1$ quantifiers.

- if $Q_1 = \exists$, then either $\phi'[x_1 \leftarrow \top]$ is valid or $\phi'[x_1 \leftarrow \bot]$ is. We handle the first case, the second one being symmetric. For a literal ℓ_i, we write L_{ℓ_i} for the set of clauses containing this literal.
  If $\phi'[x_1 \leftarrow \top]$ is valid, by induction we know that there exists a strategy σ^{x_1} such that all the players not in L_{x_1} lose from state s_2 (because $\mathcal{G}(\phi'[x_1 \leftarrow \top])$ is the same game as $\mathcal{G}(\phi)$, but starting from s_2 and without the players in L_{x_1}). We define the strategy σ such that $\sigma(s_1) = x_1$ and $\sigma(s_1 \cdot x_1 \cdot \rho) = \sigma^{x_1}(\rho)$. An outcome of σ will necessarily visit x_1, making all the players in L_{x_1} lose; because σ follows σ^{x_1}, all the other players (except A) will lose.
- if $Q_1 = \forall$, then $\phi'[x_1 \leftarrow \top]$ is valid. Using the induction hypothesis we know that from s_2 there is a strategy σ^{x_1} that makes all the players not in L_{x_1} lose. Similarly, $\phi'[x_1 \leftarrow \bot]$ is valid, so there is a strategy $\sigma^{\neg x_1}$ that makes all the players not in $L_{\neg x_1}$ lose. We define a new strategy σ as follows: $\sigma(s_1 \cdot x_1 \cdot \rho) = \sigma^{x_1}(\rho)$ and $\sigma(s_1 \cdot \neg x_1 \cdot \rho) = \sigma^{\neg x_1}(\rho)$. Consider an outcome of σ: if it visits x_1, then all the player in L_{x_1} lose, and because the path follows σ^{x_1}, the players not in L_{x_1} also lose. The other case is similar.

We now turn to the converse implication. Assume the formula is not valid. We prove that for any strategy σ of Player A, there is an outcome ρ of this strategy such that some player C_j wins.

We again proceed by induction, beginning with the case where $n = 1$.

- if $Q_1 = \exists$, then both $\phi'[x_1 \leftarrow \top]$ and $\phi'[x_1 \leftarrow \bot]$ are false. This entails that one of the clauses is $\bot$ (no non-empty disjunction is always false), and the corresponding safety condition is empty. The corresponding player will not lose, whatever Player A does.
- if $Q_1 = \forall$, then one of $\phi'[x_1 \leftarrow \top]$ and $\phi'[x_1 \leftarrow \bot]$ is false. If both are false, then we are back to the previous case. Otherwise, one of the clauses contains x_1 and another one contains $\neg x_1$. Then along any run, one of the players C_j wins, as either x_1 or $\neg x_1$ will be visited.

Now, assuming that the result holds for formulas with n quantifiers, we prove the result with $n+1$ quantifiers.

- if $Q_1 = \exists$, then both $\phi'[x_1 \leftarrow \top]$ and $\phi'[x_1 \leftarrow \bot]$ are false; using the induction hypothesis one of the players wins.
- if $Q_1 = \forall$, $\phi'[x_1 \leftarrow \top]$ is not valid or $\phi'[x_1 \leftarrow \bot]$ is not valid. We will treat the first case and the second one is done symmetrically.
  By induction hypothesis in the game $\mathcal{G}(\phi'[x_1 \leftarrow \top])$ there is a path ρ' following σ such that not all the players not in L_{x_1} are losing. We choose $\rho = s_1 \cdot x_1 \cdot \rho'$, only the players in L_{x_1} are losing along ρ', so at least one of the players is not losing. ◀

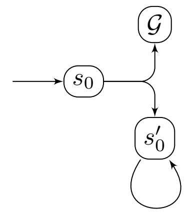

Figure 11 $\mathcal{G}'(\phi)$

◀

◀

This result can be extended to the existence problem.

▶ **Proposition.** *The existence problem for finite safety games is* PSPACE*-hard.*

Proof. We adapt the previous construction by adding a new state s_0, with a nondeterministic transition to s_1 (the initial state of $\mathcal{G}(\phi)$) and to another new state s'_0, which is equipped with a self-loop. State s'_0 is unsafe (i.e., losing) for all the players C_j. In other terms, for the path that loops in s'_0, only Player A wins.

If ϕ is valid, then in $\mathcal{G}(\phi)$ there is a strategy for Player A under which all the players lose. This strategy (extended to begin from s_0) together with the outcome that loops in s'_0 form a pseudo-Nash equilibrium, in which Player A wins.

If ϕ is not valid, then for any strategy of Player A in $\mathcal{G}(\phi)$, there is an outcome where some player C_j wins. Thus for any Player-A strategy, there is an outcome where some player C_j wins and Player A loses, and an outcome where all players C_j lose and Player A wins. Hence there exists no pseudo-Nash equilibrium. ◀ ◀

C.3 Proof of Theorem 24

▶ **Theorem 24.** *The verification, existence, and constrained existence problems are* NP*-complete for finite deterministic games with objectives defined by* 1*-weak deterministic Büchi automata of bounded size.*

Again, we start by proving the upper bound.

▶ **Proposition.** *The verification, existence, and constrained existence problems are in* NP *for finite deterministic games with objectives defined by* 1*-weak deterministic Büchi automata of bounded size.*

Proof. We first need a auxiliary procedure $g(P, n, q, R)$ which is a modification of f where we keep the result we already computed in a table R, so that we do not compute twice the repellor for the same parameters. Since we do not compute the same result twice, the number of computation is bounded by $2^P \times n \times \prod_{A_i \in P} |Q_i|$; this is exponential but we will only call this procedure with very small (logarithmic) parameters.

The non-deterministic procedure proceeds by guessing a set of losers L, a sequence of moves $(m_{\mathrm{Agt}})_{1 \leq k \leq i+j}$ describing an infinite path in the graph. We can choose for this sequence, a path of length i shorter than $|\mathrm{States}| \times \sum_{A_i \in \mathrm{Agt}} |Q_i|$, plus a loop of length j shorter than $|\mathrm{States}| \times |\mathrm{Agt}|$, this is enough to satisfy the winning conditions of the winners.

Algorithm 2 Procedure $g(P, n, q, R)$

```
if $R[P,n,q] \neq \bot$ then
  return $R$
end if
if $n = 0$ then
  $R[P,n,q] \leftarrow \varnothing$
  return $R$
end if
if $P = \varnothing$ then
  $R[P,n,q] \leftarrow \text{States}$
  return $R$
end if
$S' \leftarrow \text{States}$
repeat
  $S \leftarrow S'$
  for all $s \in S$ do
    found_m $= \bot$
    for all $m_{\text{Agt}} \in \text{Mov}(s)$ do
      found_s' $= \bot$
      for all $s' \in \text{States}$ do
        $P' \leftarrow \text{Susp}(s, s', m_{\text{Agt}})$
        $R \leftarrow g(P \cap P', n-1, q \xrightarrow{s}, R)$
        if $P \not\subseteq P'$ or $q \neq q \xrightarrow{s}$ then
          $R \leftarrow g(P' \cap P, n, q \xrightarrow{s}, R)$
        end if
        if $\exists B \in P \cap P'.\ (s', q \xrightarrow{s}) \in \Omega_B$ and $s' \notin R[P \cap P', n-1, q \xrightarrow{s}]$ then
          found_s' $= \top$
          break
        end if
        if $\left( \begin{array}{l} ((P \subseteq P' \text{ and } q = q \xrightarrow{s}) \text{ and } s' \notin S') \\ \text{ or} \\ ((P \not\subseteq P' \text{ or } q \neq q \xrightarrow{s}) \text{ and } s' \notin R[P' \cap P, n, q \xrightarrow{s}]) \end{array} \right)$ then
          found_s' $= \top$
          break
        end if
      end for
      if $\neg$found_s' then
        found_m $= \top$
        break
      end if
    end for
  end for
  if $\neg$found_m then
    $S' \leftarrow S' \setminus \{s\}$
  end if
until $S' = S$
$R \leftarrow S'$
return $R$
```

We first check that the path does not satisfy the winning conditions of the losers, and satisfies the winners. Then, along this path ρ, we look at all the possible deviations, we compute the repellor using the procedure $g(P, n, q, R)$ for the set of players: $\{\mathrm{Susp}(\rho_{\leq k}, s') \mid s' \in \mathrm{States}, 0 \leq k \leq i + j\}$.

When the game is deterministic, the size of the set of suspect for a deviation is small, that will be the reason why the call to g will return in polynomial time. Let m be the maximum size of this sets of suspect, because the game is deterministic, each suspect must have two different actions, therefore $2^m \leq |\mathrm{Tab}|$. The procedure g return in time $t = 2^P \times n \times \prod_{A_i \in P} |Q_i|$. We have that $2^P \leq 2^m$ and $\prod_{A_i \in P} |Q_i| \leq (\max_{A_i \in P} |Q_i|)^P \leq |\mathrm{Tab}|^{\log_2 \max_{A_i \in P} |Q_i|}$. We can now bound t by $n \times |\mathrm{Tab}|^{1+\log_2 \max_{A_i \in P} |Q_i|}$.

Therefore we can bound the number of computation our non deterministic procedure we will have to perform by $|\mathrm{States}| \times (|\mathrm{Agt}| + 1) \times |\mathrm{States}| \times |\mathrm{Tab}|^{1+\log_2 \max_{A_i \in P} |Q_i|}$. This is polynomial in the case were the size of each Q_i is bounded. ◀

Algorithm 3 Procedure computing Nash-equilibrium for deterministic games

guess $\nu \in \{0; 1\}^{\mathrm{Agt}}$
guess $i \leq |\mathrm{States}| \times \sum_{A_i \in \mathrm{Agt}} |Q_i|$ and $j \leq |\mathrm{States}|$
guess $M_k \in \mathrm{Act}^{\mathrm{Agt} \times (i+j+1)}$
$\rho_0 \leftarrow s$
$q_0 \leftarrow \prod_{A \in \mathrm{Agt}} q_{A,0}$
for all $k \in [1; i + j + 1]$ **do**
  $\rho_k \leftarrow \mathrm{Tab}(\rho_{k-1}, M_k)$
  $q_k \leftarrow \prod_{A \in \mathrm{Agt}} \delta_A(q_{k-1}, \rho_{k-1})$
  for all $s' \in \mathrm{States} \setminus \{\rho_k\}$ **do**
    $P' \leftarrow \mathrm{Susp}((\rho_{k-1}, s'), M_{k-1})$
    $n \leftarrow |\mathrm{States}| \times \sum_{i \in \mathrm{Agt}} |Q_i| \times |P'|$
    if $s' \notin g(P', n, q_k, (\bot)_{P \subseteq P', 0 \leq i \leq n, q \in \prod Q_i})$ **then**
      return false
    end if
  end for
end for
if $\rho_{i+j+1} \neq \rho_i$ or $q_{i+j+1} \neq q_i$ **then**
  return false
end if
for all $A \in \mathrm{Agt}$ **do**
  if $q_{i+j+1,A} \in F_A \not\Leftrightarrow \nu_A = 1$ **then**
    return false
  end if
end for
return true

◀

The lower bound holds even for turn-based deterministic games.

▶ **Proposition.** *The verification problem for finite turn-based deterministic safety games is* NP-*hard.*

Proof. The proof directly follows from the PSPACE-hardness proof: here we do not have

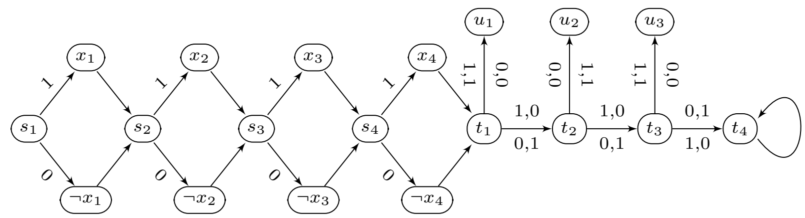

Figure 12 Example of the construction for 4 variables and 3 clauses

nondeterministic nodes, so that we can only encode existential quantification, which yields an encoding of SAT.

◀

The result can be strengthened to the existence problem if we allow non-turn-based games.

▶ **Proposition.** *The existence problem for finite deterministic safety games is* NP*-hard.*

Proof. We extend the previous construction by adding one state t_j for each clauses C_j, and one final state. For each $j \in [1,m]$, we plug a module M_j^C with one extra state u_j and two edges (t_j,t_{j+1}) and (t_j,u_j). In state t_j, Player C_j plays the matching-penny game with A_1. Player A_1 will want to avoid all the u_j states, while Player C_j wants to avoid t_{m+1}.

If the formula ϕ is valid then there is a path in $\mathcal{G}$ make all the player loose, then we extend this path to a path to t_{m+1}, and then, no player C can improve her strategy, and A is winning.

If the formula ϕ is not valid, consider any path in $\mathcal{G}'$:

- if it ends in t_{m+1}, then one of the players C_j was not losing in the game $\mathcal{G}$, therefore she can improve her outcome by switching her strategy in state t_j;
- if it ends in one of the u_j, then A is losing and she can improve by switching to a strategy match the choice of player C_j in state t_j for all $1 \leq j \leq m$.

Therefore, there is an equilibrium if and only if the formula is satisfiable. ◀ ◀

D Appendix for Section 6

D.1 Proof of Theorem 26

▶ **Theorem 26.** *The constrained existence problem is in* NP *for finite deterministic concurrent games with parity objectives if we assume that strategies can observe actions.*

The proof resembles a proof in [A7] for concurrent limit-average games. For the proof, we assume that histories $\mathrm{Hist}_{\mathcal{G}}$ are finite sequences $s_0 m^1_{\mathrm{Agt}} s_1 \ldots m^p_{\mathrm{Agt}} s_p \ldots$ where for every i, $s_i \in S$ and $m^{i+1}_{\mathrm{Agt}} \in \mathrm{Act}^{\mathrm{Agt}}$ is a legal move at s_i. The rest of the vocabulary used is lifted to this new framework.

Given a strategy profile, if one of the players deviates, then all the other players will be aware of her deviation (using previous notions, the set of suspect will be reduced to a single player) and together (all players but the betrayer) will be able to collaborate and punish her.

Let $\mathcal{G} = (\mathrm{States}, \mathrm{Edg}, \mathrm{Agt}, \mathrm{Act}, \mathrm{Mov}, \mathrm{Tab}, (\mathcal{L}_A)_{A\in\mathrm{Agt}})$ be a finite deterministic concurrent game. For every $B \in \mathrm{Agt}$, we define

$$\mathrm{Rep}(B) = \{s \in \mathrm{States} \mid \text{there exists } \sigma \in \mathrm{Strat}^{\mathrm{Agt}\setminus\{B\}} \text{ such that } \rho \notin \mathcal{L}_B \text{ for all } \rho \in \mathrm{Out}(\sigma, s)\}\,.$$

▶ **Lemma 27.** *Let* $B \in \mathrm{Agt}$*. There exists a memoryless strategy profile* $\sigma \in \mathrm{Strat}^{\mathrm{Agt}\setminus\{B\}}$ *such that* $\rho \notin \mathcal{L}_B$ *for all* $\rho \in \mathrm{Out}(\sigma, s)$ *that starts in a state* $s \in \mathrm{Rep}(B)$*.*

Proof. We define a turn-based two-player zero-sum parity game $\mathcal{G}'$ with players A and B as follows: The set of states of $\mathcal{G}'$ is $\mathrm{States}' = \mathrm{States} \cup (\mathrm{States} \times \mathrm{Act}^{\mathrm{Agt}})$. At a state $s \in \mathrm{States}$, player A chooses a move m_{Agt} that is legal at s in $\mathcal{G}$, which leads the game to the state (s, m_{Agt}). At a state of the form (s, m_{Agt}), player B chooses an action $m' \in \mathrm{Mov}(s, B)$, which leads the game to the unique state s' s.t. $(s, s') \in \mathrm{Tab}(s, m_{\mathrm{Agt}}[B \mapsto m'])$. Finally, the priority of a state $s \in \mathrm{States}$ or $(s, m_{\mathrm{Agt}}) \in \mathrm{States} \times \mathrm{Act}^{\mathrm{Agt}}$ (for player A) equals the priority of s in $\mathcal{G}'$ for player B plus 1.

By [A4], there exists a set $W \subseteq \mathrm{States}'$ (the *winning region*) and memoryless strategies σ_A and σ_B for players A and B such that σ_A is winning from every state $s \in W$ and σ_B is winning from every state $s \in \mathrm{States}' \setminus W$. We can translate σ_A into a memoryless strategy profile σ for the coalition $\mathrm{Agt} \setminus \{B\}$ in $\mathcal{G}$ such that $\rho \notin \mathcal{L}_B$ for all $\rho \in \mathrm{Out}(\sigma, s)$ that starts in a state $s \in W$. Hence, $W \subseteq \mathrm{Rep}(B)$. We claim that also $\mathrm{Rep}(B) \subseteq W$, which implies that σ is the strategy profile we are looking for. Hence, let $s \in \mathrm{Rep}(B)$ and fix a strategy $\sigma \in \mathrm{Strat}^{\mathrm{Agt}\setminus\{B\}}$ such that $\rho \notin \mathcal{L}_B$ for all $\rho \in \mathrm{Out}(\sigma, s)$. But such a strategy can be extended to a strategy σ_A for player A in $\mathcal{G}'$ which is winning from s. Hence, σ_B is not winning from s which implies that $s \in W$. ◀ ◀

Roughly speaking, player B only has an incentive to deviate if the game is not inside $\mathrm{Rep}(B)$, since otherwise any deviation can be punished by the coalition $\mathrm{Agt} \setminus \{B\}$.

To formalise this, we now adapt the notion of secure move to the new framework. If $P \subseteq \mathrm{Agt}$, we say that a move $m_{\mathrm{Agt}} \in \mathrm{Act}^{\mathrm{Agt}}$ is *P-secure* at state s if for all $B \in P$ the following holds: $s' \in \mathrm{Rep}(B)$ for all $s' \in \mathrm{States}$ and $m \in \mathrm{Act}$ such that $m_{\mathrm{Agt}}[B \mapsto m]$ is legal at s and $(s, s') \in \mathrm{Tab}(s, m_{\mathrm{Agt}}[B \mapsto m])$.

Then we can define the graph $\mathcal{S}_{\mathcal{G}}(P) = (\mathrm{States}, E)$, where $(s, t) \in E$ if and only if there exists a P-secure move m_{Agt} at s such that $(s, t) \in \mathrm{Tab}(s, m_{\mathrm{Agt}})$.

We can now prove the following lemma, which precisely characterises the payoffs of Nash equilibria.

▶ **Lemma 28.** *Let* $\nu \colon \mathrm{Agt} \to \{0, 1\}$ *and* $s \in \mathrm{States}$*. There exists a Nash equilibrium from* s *in* $\mathcal{G}$ *with payoff* ν *if and only if there exists an infinite path in* $\mathcal{S}_{\mathcal{G}}(\nu^{-1}(\{0\}))$ *from* s *with payoff* ν*.*

Proof. Let σ be a Nash equilibrium from s and $\pi = (s_j)_{0 \le j < \infty} \in \mathrm{Out}(\sigma, s)$ its unique outcome. We claim that π is a path in $\mathcal{S}_{\mathcal{G}}(P)$, where $P = \nu^{-1}(\{0\})$. Otherwise, fix the minimal j such that there is no P-secure move m_{Agt} at s_j with $(s_j, s_{j+1}) \in \mathrm{Tab}(s_j, m_{\mathrm{Agt}})$. This means that some player $B \in P$ can play from s_j to a state $s \in \mathrm{States} \setminus \mathrm{Rep}(B)$ from where she can play a strategy to enforce a win for her, which contradicts the fact that σ is a Nash equilibrium.

Now, let $\pi = (s_j)_{0 \le j < \infty}$ be an infinite path in $\mathcal{S}_{\mathcal{G}}(P)$ from s with payoff ν, where $P = \nu^{-1}(\{0\})$. We define a strategy profile σ as follows: For histories of the form $h = s_0 m^0_{\mathrm{Agt}} s_1 \dots s_{k-1} m^{k-1}_{\mathrm{Agt}} s_k$, we set $\sigma(h)$ to a move m_{Agt} with $\mathrm{Tab}(s_k, m_{\mathrm{Agt}}) = \{(s_k, s_{k+1})\}$ that is P-secure at s_k. For all other histories $h = t_0 m^0_{\mathrm{Agt}} t_1 \dots t_{k-1} m^{k-1}_{\mathrm{Agt}} t_k$, consider the least j such that $s_{j+1} \ne t_{j+1}$. If m^j_{Agt} differs from a P-secure move m_{Agt} at s_j in precisely one entry B, we set $\sigma(h) = \sigma^*(t_k)$, where σ^* is a (fixed) positional strategy profile of the coalition $\mathrm{Agt} \setminus \{B\}$ such that $\rho \notin \mathcal{L}_B$ for all $\rho \in \mathrm{Out}(\sigma, s)$ with $s \in \mathrm{Rep}(B)$, which is guaranteed to exist by Lemma 27 (the action selected by player B is arbitrary); otherwise, $\sigma(h)$ can be chosen arbitrarily. It is easy to see that σ is a Nash equilibrium with induced play π. ◀ ◀

We can now describe a nondeterministic polynomial-time algorithm for the constrained existence problem: First, we guess a payoff ν that satisfies the given constraint and set $P = \nu^{-1}(0)$. Then, for each $B \in P$, we guess a memoryless strategy $\sigma \in \mathrm{Strat}^{\mathrm{Agt}\setminus\{B\}}$ and compute the set $\mathrm{Rep}'(B) = \{s \in \mathrm{States} \mid \rho \notin \mathcal{L}_B \text{ for all } \rho \in \mathrm{Out}(\sigma, s)\}$ in polynomial time. From this set, we can compute the graph $\mathcal{S}'_{\mathcal{G}}(P) = (\bigcap_{B\in P} \mathrm{Rep}'(B), E')$, which is defined like $\mathcal{S}_{\mathcal{G}}(P)$ but with $\mathrm{Rep}'(B)$ substituted for $\mathrm{Rep}(B)$. Finally, the algorithm accepts if there exists an infinite path in $\mathcal{S}'_{\mathcal{G}}(P)$ from s with payoff ν; Emerson and Lei [A5] showed that the existence of such a path can be checked in polynomial time. The correctness of the algorithm follows from Lemmas 28 and 27 as well as the fact that $\mathcal{S}'_{\mathcal{G}}(P)$ is a subgraph of $\mathcal{S}_{\mathcal{G}}(P)$ for every choice of σ.

D.2 Proof of Proposition 27

▶ **Proposition 27.** *The verification, existence and constrained existence problems are* coNP*-hard for finite deterministic concurrent games with parity objectives. In particular, unless* NP = coNP*, this problem does not belong to* NP*.*

Proof. We first prove that the verification problem is coNP-hard, which implies hardness of the constrained existence problem. We reduce from the problem whether the first player has a wining strategy in a deterministic turn-based game where the objective is a conjunction of two parity conditions; this problem was proved to be coNP-hard in [A2]. Given such a game $\mathcal{G}$, say played by Players 1 and 2, and an initial vertex v_0, we construct a concurrent three-player parity game $\mathcal{G}'$ such that Player 1 has a winning strategy in $\mathcal{G}$ from v_0 if and only if $\mathcal{G}'$ has a Nash equilibrium from v_0 with payoff $(0, 0, 0)$.

Without loss of generality, we can assume that in $\mathcal{G}$ each vertex has at most two successors (because $\mathcal{G}$ is turn-based). The game $\mathcal{G}'$ is obtained from $\mathcal{G}$ as follows: $\mathcal{G}'$ is played by Players 1, 2 and 3 and has actions a and b. If v is a vertex of $\mathcal{G}$ with successors v_1 and v_2 that is controlled by Player 2, then in $\mathcal{G}'$ the moves $\langle a, a, b\rangle$ and $\langle a, b, a\rangle$ result in the transition (v, v_1) and the moves $\langle a, a, a\rangle$ and $\langle a, b, b\rangle$ result in the transition (v, v_2), and these are the only legal moves at v. If v is controlled by Player 1, then the only legal moves at v are $\langle a, a, a\rangle$ and $\langle b, a, a\rangle$, where the move $\langle a, a, a\rangle$ results in the transition (v, v_1) and the move $\langle b, a, a\rangle$ results in the transition (v, v_2). All states of $\mathcal{G}'$ have priority 1 for Player 1. Finally, the priority function for Players 2 and 3 are obtained from the priority functions in $\mathcal{G}$ by adding 1 to each priority.

Clearly, if σ_1 is a winning strategy in $\mathcal{G}$ from v_0, then $\langle \sigma_1, \sigma_2, \sigma_3\rangle$ is a Nash equilibrium of $\mathcal{G}'$ from v_0 with payoff $(0, 0, 0)$ for every pair of strategies $\langle \sigma_2, \sigma_3\rangle$ for Players 2 and 3. On the other hand, let $\langle \sigma_1, \sigma_2, \sigma_3\rangle$ be a Nash equilibrium of $\mathcal{G}'$ from v_0 (in state-based strategies) with payoff $(0, 0, 0)$. We claim that σ_1 is a winning strategy in $\mathcal{G}$ from v_0. Otherwise, there would exist a play π of $\mathcal{G}$ from v_0, compatible with σ_1, that fulfils one of the parity conditions; without loss of generality, assume that π fulfils the first parity condition. Then, π is winning for Player 2 in $\mathcal{G}'$. Moreover, by the definition of $\mathcal{G}'$, there exists a (state-based) strategy σ'_2 for Player 2 such that $\pi \in \mathrm{Out}(v_0, \langle \sigma_1, \sigma'_2, \sigma_3\rangle)$. Hence, $\langle \sigma_1, \sigma_2, \sigma_3\rangle$ is not a Nash equilibrium from v_0, a contradiction.

Finally, we show how to adapt our construction to the existence problem: to the game $\mathcal{G}'$ described above, we add five new states s_0, s_1, s_2, s_3 and s_4. The states s_2, s_3 and s_4 are sink states which are winning for Players 1 and 2, Players 1 and 3, and Player 1, respectively. In s_0, the moves $\langle a, a, a\rangle$ and $\langle a, b, b\rangle$ result in going to s_2; the moves $\langle a, a, b\rangle$ and $\langle a, b, a\rangle$ result in going to s_3, and all other moves (i.e. all moves where Player 1 plays action b) result in going to s_1. Finally, in s_1 the moves $\langle a, a, b\rangle$, $\langle a, b, a\rangle$, $\langle b, a, b\rangle$ and $\langle b, b, a\rangle$ result in going to v'_0, and all other moves result in going to s_4. It is easy to see that any Nash equilibrium of $\mathcal{G}'$ from v'_0 with payoff $(0,0,0)$ induces a Nash equilibrium from s_0 that goes to s_4 and has payoff $(1,0,0)$. On the other hand, if σ is a Nash equilibrium of $\mathcal{G}'$ from s_0, then it must give payoff 1 to Player 1 because playing action a in s_0 ensures a winning play for Player 1. If the equilibrium play lead to s_2, then Player 3 could improve her payoff by switching her action at s_0. Analogously, if the equilibrium play lead to s_3, then Player 2 could improve her payoff. Hence, the equilibrium play must lead to s_4 and have payoff $(1,0,0)$. But then $\mathcal{G}'$ must have an equilibrium from v'_0 with payoff $(0,0,0)$ since otherwise Player 2 or 3 could improve her payoff by switching her action at s_1 and playing a suitable strategy from v'_0. ◀ ◀

References

[A1] P. Bouyer, R. Brenguier, and N. Markey. Nash equilibria for reachability objectives in multiplayer timed games. Research Report LSV-10-12, ENS Cachan, France, 2010.

[A2] K. Chatterjee, T. A. Henzinger, and N. Piterman. Generalized parity games. In FoSSaCS'07, LNCS 4423, p. 153–167. Springer, 2007.

[A3] K. Chatterjee, R. Majumdar, and M. Jurdziński. On Nash equilibria in stochastic games. In CSL'04, LNCS 3210, p. 26–40. Springer, 2004.

[A4] E. A. Emerson and C. S. Jutla. Tree automata, mu-calculus and determinacy (extended abstract). In FOCS'91, p. 368–377. IEEE, 1991.

[A5] E. A. Emerson and C.-L. Lei. Modalities for model checking: branching time logic strikes back. *Science of Computer Programming*, 8(3):275–306, 1987.

[A6] R. E. Tarjan. Depth-first search and linear graph algorithms. *SIAM J. Computing*, 1(2):146–160, 1972.

[A7] M. Ummels and D. Wojtczak. The complexity of Nash equilibria in limit-average games. In CONCUR'11, LNCS 6901, p. 482–496. Springer, 2011.